공학박사학위논문

Online Advertising Assignment Problems Considering Realistic Constraints

현실제약을 고려한 온라인 광고 할당 문제

2020 년 8 월

서울대학교 대학원

산업공학과

김 광

Online Advertising Assignment Problems Considering Realistic Constraints

현실제약을 고려한 온라인 광고 할당 문제

지도교수 문 일 경

이 논문을 공학박사 학위논문으로 제출함

2020 년 5 월

서울대학교 대학원

산업공학과

김 광

김 광의 공학박사 학위논문을 인준함

2020 년 6 월

위 원 장	이 경 식	(인)
부위원장	문 일 경	(인)
위 원	박 건 수	(인)
위 원	이 영 훈	(인)
위 원	정 태 수	(인)

Abstract

Online Advertising Assignment Problems Considering Realistic Constraints

Gwang Kim
Department of Industrial Engineering
The Graduate School
Seoul National University

With a drastic increase in online communities, many companies have been paying attention to online advertising. The main advantages of online advertising are traceability, cost-effectiveness, reachability, and interactivity. The benefits facilitate the continuous popularity of online advertising. For Internet-based companies, a well-constructed online advertisement assignment increases their revenue. Hence, the managers need to develop their decision-making processes for assigning online advertisements on their website so that their revenue is maximized.

In this dissertation, we consider online advertising assignment problems considering realistic constraints. There are three types of online advertising assignment problems: (i) Display ads problem in adversarial order, (ii) Display ads problem in probabilistic order, and (iii) Online banner advertisement scheduling for advertising effectiveness. Unlike previous assignment problems, the problems are pragmatic approaches that reflect realistic constraints and advertising effectiveness. Moreover,

the algorithms the dissertation designs offer important insights into the online advertisement assignment problem.

We give a brief explanation of the fundamental methodologies to solve the online advertising assignment problems in Chapter 1. At the end of this chapter, the contributions and outline of the dissertation are also presented. In Chapter 2, we propose the display ads problem in adversarial order. Deterministic algorithms with worst-case guarantees are designed, and the competitive ratios of them are presented. Upper bounds for the problem are also proved. We investigate the display ads problem in probabilistic order in Chapter 3. This chapter presents stochastic online algorithms with scenario-based stochastic programming and Benders decomposition for two probabilistic order models. In Chapter 4, an online banner advertisement scheduling model for advertising effectiveness is designed. We also present the solution methodologies used to obtain valid lower and upper bounds of the model efficiently. Chapter 5 offers conclusions and suggestion for future studies.

The approaches to solving the problems are meaningful in both academic and industrial areas. We validate these approaches can solve the problems efficiently and effectively by conducting computational experiments. The models and solution methodologies are expected to be convenient and beneficial when managers at Internet-based companies place online advertisements on their websites.

Keywords: Online advertising assignment, Display ads, Online banner advertisement scheduling, Online algorithm, Stochastic programming, Advertising effectiveness

Student Number: 2014 - 22636

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Online advertising is a method for advertising with the Internet to promote products or services to customers. It has become a significant source of income for many Internet-based companies. Online advertising has advantages of traceability, cost-effectiveness, reachability, and interactivity compared to other promotional methods [95]. According to an Internet Advertising Bureau (IAB) report [68], online advertising revenue in the United States was $107.5 billion in 2018.

In online advertising market, there are two participants: advertisers and publishers. Advertisers request their advertisements to be displayed on websites and publishers place requested advertisements on their websites. Online advertising assignment is generally decided through contracts [99]. Advertisers and publishers achieve contracts by using compensation methods. Most of the Internet-base companies, including *Google AdWords* and *Naver Click Choice*, use cost-per-click (CPC) method based on customer performance as a compensation method between them [28]. It means advertisers deposit their budget to an agency, and then publishers gain revenue from the deposit each time a user clicks on their ads. Publishers need to decide appropriate advertisements to be displayed to increase the number of clicks from the users. For this reason, research on publishers' decision making related to online advertising has been studied [95, 15, 99, 48, 49, 40, 82, 59, 37, 36, 35, 18]. Likewise, in this dissertation, we cover online advertising assignment problems that publishers solve.

There are several ad formats (channels) in online advertising. Among them, *search engine advertising* and *display advertising* are the most widely used formats. In search engine advertising, advertisers want to show their ads to website users who search for related keywords. If advertisers have a limited budget or want to sell a specific product or service, search engine advertising can be a good fit. Publishers should select an ad at the moment a user searches for related keywords. They select an ad by considering the relation between the user and keyword. On the other hand, display advertising is different from search engine advertising that selects ads based on keywords. Advertisers want to display their graphical ads (e.g., banner ads) on the specific space positioned on either side, top, or bottom of a web page. If advertisers want to build brand awareness or do not have an immediate sale product, display advertising can be a good fit.

In this dissertation, we cover two online advertising assignment problems that model search engine advertising and display advertising, respectively. The two problems are the *display ads problem* and the *online banner advertisement scheduling problem*. In Chapters 2 and 3, we present the display ads problem that can be represented as search engine advertising. In Chapter 4, we present the online banner advertisement scheduling problem that can be represented as display advertising. We give a brief explanation of the two problems in the next two sections. At the end of this chapter, the contributions and outline of the dissertation are presented.

1.1 Display Ads Problem

The display ads problem, which can be represented as search engine advertising, has been studied both theoretically and practically [86, 49, 14, 83, 50]. When a user searches for a related keyword, publishers select an ad to be displayed among the set of eligible ads. Because traffic to the website is not known in advance, the problem should be handled in real time. The display ads problem aims to select an appropriate ad in real time to increase the probability that a user clicks the ad when he or she arrives. Because publishers select an ad in real time without knowing the information of the future, we deal with the problem in terms of online optimization. To solve the problem related to *online optimization*, this dissertation presents *online algorithms*. We give a brief explanation of online optimization and online algorithm in Section 1.1.1.

The display ads problem has two features. First, all eligible ads have different weights. The weight might be a prediction of click-through probability or an estimate of targeting quality [16, 49]. The weights are revealed when a user arrives. Publishers select an ad after considering the weights of the eligible ads. The value of weight can be predicted through linear models, hybrid methods, or deep learning [4, 27, 75, 123]. However, in this dissertation, we do not cover the real-time prediction on the weight and assume to get to know the weight when a user arrives. Second, each advertiser has its budget deposited through which publishers gain revenue when its ad is clicked in a specific period. Because of the limited budget, each ad has the maximum number of displays in a specific period. Another reason why the problem considers the maximum number of displays is to give the chance of displaying their ads to as many advertisers as possible. In fact, it is reported that large publishers usually can display only 60% of the eligible ads because too many advertisers contract and deposit their budget to an agency [53, 37, 49].

1.1.1 Online Algorithm

Online algorithms have recently been applied in fields of computer science and operations research. In practice, optimization problems such as scheduling, resource allocation may be faced with the situation in which there is no or incomplete information of the future. We call this situation *online*. Online optimization is an optimization approach that solves optimization problems having no or incomplete knowledge of the future. The problems in online situations are different from the classical problems in which all input information is given (*offline*) [72, 51].

To deal with the problems in online situation, we present online algorithms. An online algorithm, which is designed to solve online problems, is the one that can process its input piece-by-piece in a serial fashion without having the entire knowledge of inputs available from the beginning [80, 10]. It means that input is revealed to the algorithm incrementally, then output is produced incrementally according to the revealed input. On the other hand, an offline algorithm solves the problem with knowing the whole input data in advance. Offline algorithms can find the optimal solutions by using the complete information on input. Meanwhile, online algorithms are more likely to find solutions that are not optimal because the information of the input is incomplete.

In online problems, competitive analysis is used to analyze the online algorithms. The competitive ratio in this analysis is introduced to measure the performance of the online algorithm, and defined as the worst-case ratio between the performance of an online algorithm and an optimal offline algorithm. An online algorithm is called c-competitive if the cost (or profit) of the online algorithm is never worse than c times the cost (or profit) of the optimal offline algorithm. In other words, given an online problem, let ALG and OPT be the value of the objective function by the online algorithm and the optimal value, respectively. For any input data (sequence), the online algorithm is c-competitive if $ALG/OPT \geq c$ is satisfied.

Recent literature on online algorithms has been trying to design new online algorithms that have a tighter competitive ratio compared to that of the previous algorithm. In addition to deterministic algorithms, randomized algorithms have been devised to obtain better competitive ratios in online problems. If we prove that no deterministic (or randomized) algorithms can have a tighter competitive ratio than that of the online algorithm, the algorithm is called an optimal online algorithm. One of the objectives for research on online problems is to get a (near)-optimal online algorithms in the online problems. For example, $RANKING$ algorithm is known as an optimal online algorithm in a online bipartite matching problem [99, 81]. In Chapters 2 and 3, we deal with online algorithms in the display ads problem, which is one of the online optimization problems.

1.2 Online Banner Advertisement Scheduling Problem

The online banner advertisement scheduling problem,which can be represented as display advertising, has been studied both theoretically and practically [1, 2, 7, 36, 54, 74]. Publishers have a set of the eligible ads, through contracts with advertisers, to be displayed on the site in the next planning period. In reality, a decision on banner advertisement assignment is conducted half-a-day or a full day in advance [37]. Therefore, contrary to the display ads problem, publishers decide to assign banner ads on slots of the next period while knowing all information on the eligible ads in the online banner advertisement scheduling problem.

The online banner advertisement scheduling problem aims to assign appropriate banner ads on slots to increase the number of times users click the ads. The problem has two features. First, all banner ads have different sizes (heights). Without exceeding the height of the slot, several banner ads can be displayed on the slot simultaneously. Second, we know that publishers can jointly select several ads to display on the slot. The advertising effectiveness is different depending on the set of banner ads displayed. In the past, publishers focused on space utilization of the slots because the bid was different depending on the space size of the ad displayed [37, 59, 95]. After developing compensation methods like CPC, research on the problem has changed from maximizing space utilization to maximizing advertising effectiveness (e.g., maximizing the number of times users click the ad).

1.3 Research Motivations and Contributions

First of all, we discuss the motivations for each problem in this section. The problems in this dissertation cover online advertising assignment problems considering realistic constraints or situations. First, for the display ads problem in adversarial order, the problem is represented as the edge-weighted bipartite matching problem. However, information on weights between edges is not known in advance. Only when a node on the right-hand side (referred to as a slot) arrives, the edges and weights incident of the node are revealed. It means that the online advertising assignment progresses in the real-time environment. Therefore, we cover this problem as an extension of online bipartite matching problems.

Second, the display ads problem in probabilistic order tends to be a practical perspective. Although information on weights between edges is revealed online, the company can stochastically estimate the input sequence by using historical data in real problems. Therefore, we design online algorithms that consider the information of the estimated input sequence. The algorithms should be analyzed in terms of effectiveness and efficiency.

Third, the online banner advertisement scheduling is an extension to a MAXSPACE problem of banner advertisement scheduling [1], which only focused on maximizing the space utilization of time slots. However, maximizing advertising effectiveness may be more important than space utilization. We consider four factors that influence the tendency for online users to click on the advertisement. It is known that some factors are positive for advertising effectiveness, others are negative. The problem considering the positive or negative factors at the same time has not yet been studied in online banner advertisement scheduling.

The principal contributions of the dissertation are summarized as follows:

1. For the display ads problem in adversarial order,

    - The problem can be represented as a generalization of the edge-weighted and capacitated online bipartite matching problem.

    - Considering the strict capacity constraint, deterministic algorithms with worst-case guarantees are designed. We also prove upper bounds on the competitive ratio of any deterministic algorithms.

- The gaps between the competitive ratios and upper bounds are analyzed in many cases. We derive that the deterministic algorithm may be a near-optimal algorithm according to the capacity and weight range.

2. For the display ads problem in probabilistic order,

    - We derive stochastic formulations for the problem by considering two probabilistic order models (known IID and random permutation), which are suitable for the realistic situation. The probabilistic orders are well known to be used as stochastic input models of the online matching problem.

    - The stochastic online algorithm with scenario-based stochastic programming and Benders decomposition is proposed to solve the problem.

    - We discuss the efficiency and effectiveness of the stochastic online algorithm through numerical experiments.

3. For the online banner advertisement scheduling,

    - We propose an online banner advertisement scheduling model to maximize advertising effectiveness. Advertising effectiveness is represented as the expected click-through rate (CTR) function.

    - The expected CTR function considers four factors that influence the tendency for online users to click on the advertisement. In particular, the degree to competition has not yet been discussed extensively in the literature.

- The heuristic approach using the properties of the objective function provides competitive solutions efficiently, even for large data sets of the model which is non-convex and non-linear.

1.4 Outline of the Dissertation

In this dissertation, we consider three types of online advertising assignment problems and introduce the three problems in each chapter. In Chapter 2, we study the display ads problem in the adversarial order. We present deterministic algorithms and theorems for the problem in the adversarial order. In addition, upper bounds on the competitive ratio of any deterministic algorithms are derived. In Chapter 3, we define the display ads problem in the probabilistic order. We introduce stochastic online algorithms with scenario-based stochastic programming and Benders decomposition for the probabilistic order. The contents of Chapter 2 are based on a theoretical perspective, while Chapter 3 tends to be a practical perspective. Table 1.1 shows the comparison between this dissertation and previous research on the display ads problem in terms of constraints and approaches.

Table 1.1: Comparisons of this dissertation (Ch.2 and Ch.3) and previous studies

Author (year)	**Edge-weighted**	**Capacity**	**Approach**
Feldman et al. (2009) [49]	Unequal	N(w[1])	T[3]
Korula and Pál (2009) [86]	Unequal	1	T
Haeupler et al. (2012) [61]	Unequal	1	T/P[4]
Bhalgat et al. (2012) [14]	Unequal	N(w)	T/P
Kesselheim et al. (2013) [83]	Unequal	1	T
Jaillet et al. (2014) [71]	Equal	1	T/P
Chen et al. (2015) [26]	Unequal	1	T/P
Ting et al. (2015) [118]	Unequal	N(w/o[2])	T
Bhaskar et al. (2016) [15]	Unequal	N(w/o)	T
Sun et al. (2017) [117]	Unequal	1	T
Huang et al. (2018) [65]	Unequal	1	T/P
This dissertation	Unequal	N(w/o)	T(Ch.2)/P(Ch.3)

1) w indicates ‘with free disposal assumption.’

2) w/o indicates ‘without free disposal assumption.’

3) T indicates a theoretical approach.

4) P indicates a practical approach.

In Chapter 4, we present a mathematical formulation of the online banner advertisement scheduling for advertising effectiveness. The problem considers four factors that influence the tendency for online users to click on the advertisement. The four factors are size, exposure, involvement, and competition. Table 1.2 shows the comparison between this dissertation and previous research on the online banner advertisement scheduling problem in terms of the four factors. Then, we present an expected objective function that considers these factors. The value of the expected function can be interpreted as advertising effectiveness. We also derive competitive lower and upper bounds of the problem and propose the solution methodologies used to obtain the bounds of the problem efficiently. Finally, we present our conclusions and future research directions in Chapter 5.

Table 1.2: Comparisons of this dissertation (Ch.4) and previous studies

Author (year)	**Size**	**Exposure**	**Involvement**	**Competition**
Adler et al. (2002) [1]	Yes	No	No	No
Dawande et al. (2003) [34]	Yes	No	No	No
Freund and Naor (2004) [52]	Yes	Yes	No	No
Amiri and Menon (2006) [11]	Yes	Yes	No	No
Kumar et al. (2007) [87]	Yes	Yes	Yes	No
Deane and Pathak (2009) [38]	No	Yes	Yes	No
Boskamp et al. (2011) [18]	Yes (2-D[1])	No	No	No
Deane (2012) [35]	Yes	No	Yes	No
Manik et al. (2016) [95]	Yes	Yes	Yes	No
Kaul et al. (2018) [82]	Yes (2-D)	No	Yes	No
Purnamawati et al. (2018) [108]	Yes	Yes	Yes	No
Gamzu and Koutsopoulos (2018) [54]	No	Yes	No	Yes
This dissertation	Yes	Yes	Yes	Yes

1) 2-D indicates two-dimensional banners.

Chapter 2

Online Advertising Assignment Problem in Adversarial Order

2.1 Problem Description and Literature Review

With the increase of online users, online advertising has become a significant source of income for many Internet-based companies. According to an Internet Advertising Bureau (IAB) report [68], Internet advertising revenue in the United States was $107.5 billion in 2018. The main advantages of online advertising are traceability, cost-effectiveness, reachability, and interactivity. They facilitate the continuous popularity of online advertising [95]. For these reasons, publishers who place online advertisements on their websites need to develop decision-making processes to maximize revenue. One such process involves rapidly selecting an appropriate advertisement from among those in a set of available advertisements, and then assigning it on the website, a placement defined as *a slot* [58].

An online advertising assignment problem that publishers solve corresponds to a bipartite matching problem in graph theory. The problem can be interpreted as finding an optimal matching among the advertisements and the slots because the assignment of advertisements is generally decided either by auction or through contracts [99]. Unlike the matching problem in which cardinality is maximized, the objective of the online advertising assignment problem is to find connections, through which revenue is maximized, between the advertisements and the slots. Each edge (connection) has a weight. The weight of the edge might be a prediction of click-through probability, an estimate of targeting quality, or a bid submitted by the advertiser [16, 49]. When an advertisement is assigned to a slot, the weight corresponding to the edge is realized.

In reality, information on weights is not known beforehand, making the problem uncertain. Because of this, publishers focus on the online version of the problem [46, 84, 109]. In other words, we define a bipartite graph for which information about the nodes on the left-hand side is known in advance, and the nodes on the right-hand side arrive online (one node at a time). The nodes on the left-hand side represent advertisements, and those on the right-hand side represent slots. When a node on the right-hand side arrives, the edges and weights incident of the node are revealed. An online algorithm of the problem selects one of the edges (an advertisement is displayed on the slot) or discards them (no advertisement is displayed on the slot). The decision is irrevocable [2, 17, 24, 26, 29, 50, 57, 61, 65, 71, 77, 78, 81, 94, 96, 105, 119]. Online algorithms must complete each request of the assignment without knowing the future sequence of the nodes on the right-hand side [63].

If each node on the left-hand side has an integer capacity (the maximum number of right-hand nodes being matched to the left-hand node), we call the situation a *Display ads* problem. The problem is a generalization of the edge-weighted and capacitated online bipartite matching problem. [49] gave a $(1 - \frac{1}{e})$-competitive algorithm for the display ads problem in the adversarial order when the value of each capacity is big. [48] and [14] implemented fairness and smooth delivery constraints for the display ads problem. For edge-weighted bipartite matching, [118] proposed a near-optimal algorithm for the edge-weighted b-matching problem. [117] proposed a randomized algorithm, which gives a near-optimal solution.

In the display ads problem, [49] introduced the property (assumption) referred to as free disposal. The definition of the free disposal assumption is that each node on the left-hand side is allowed to be matched more times than its capacity, (c), but the publishers gain only for the 'c' highest weights matched. In other words, the assumption allows for violating the capacity constraint. Previous research introduced the assumption in the display ads problem to obtain bounded competitive ratios in the adversarial order [14, 48, 49].

Although the problem is tractable when the assumption is allowed, the problem situation might be restricted. If some advertisers are sensitive to the number of times their advertisements are displayed, the solutions with the assumption might cause issues with trust. That is because there is a possibility that an advertisement can be displayed more times than its capacity, while other advertisements miss the chance to be displayed. Also, the display ads problem that allows for the free disposal assumption is challenging to apply to other types of problems (e.g., scheduling or resource allocation) in which resources, such as humans and machines, are strictly limited. In this study, the objective of this problem is to maximize the total weight of edges matched while considering the strict capacity constraint.

The *Adwords* problem is similar to the display ads problem in terms of the application of online bipartite matching problems. The adwords problem has an individual budget instead of an integer capacity for each node on the left-hand side, and the objective of the problem is to maximize the total budget spent [4, 15, 19, 25, 40, 41, 57, 70, 76, 90, 100, 101]. [15] proposed the adwords problem without a small-bid assumption, which is a relaxed capacity constraint. Like [15], we propose the display ads problem that does not allow the free disposal assumption. We call it the display ads problem without free disposal. The objective of this problem is to maximize the total weight of edges matched while considering the strict capacity constraint. For the adversarial order, deterministic algorithms with worst-case guarantees are designed, and the competitive ratios of them are proved. Upper bounds for the problem are also proposed.

The rest of this section is organized as follows. In Section 2.2, we define the display ads problem in the adversarial order. Section 2.3 presents deterministic algorithms and theorems for the problem in the adversarial order. In Section 2.4, upper bounds on the competitive ratio of any deterministic algorithms are presented. We also compare the gaps between the competitive ratios and the upper bounds. We summarize this section in Section 2.5.

2.2 Display Ads Problem in Adversarial Order

The display ads problem is defined as follows [49, 99, 117, 118]: For an edge-weighted bipartite graph $\boldsymbol{G} = (A, T, E, w)$, A is a set of advertisements (left); T is a set of slots (right); E is a set of edges of graph $\boldsymbol{G}$; and w is a set of weights for E. We know the information on A in advance and each advertisement $i \in A$ has capacity C_i, which is the maximum number of being matched to T. However, we do not know any information of T, E, and w, except $|T|$. The set of T arrives online, one node at a time. When a node $j \in T$ arrives, all edges incident to j as well as the weights, w_{ij}, of each are revealed. The algorithm matches a connection between a node j and one of the advertisements available or leaves the node unmatched. The decision made is irrevocable. To resolve the absence of non-trivial competitive ratios, we assume that the online algorithms know the range of the weights $[L_i, U_i]$ for each ad i.

A mathematical formulation for the display ads problem is as follows:

$$\max \sum_{i=1}^{|A|} \sum_{j=1}^{|T|} w_{ij} x_{ij} \tag{2.1}$$

$$\text{s.t.} \sum_{i=1}^{|A|} x_{ij} \leq 1 \ \forall j \in T \tag{2.2}$$

$$\sum_{j=1}^{|T|} x_{ij} \leq C_i \ \forall i \in A \tag{2.3}$$

$$x_{ij} \in \{0, 1\} \ \forall i \in A, j \in T \tag{2.4}$$

The binary decision variable, x_{ij}, is 1 if ad $i \in A$ is matched to slot $j \in T$; 0 otherwise. The objective function (2.1) maximizes the total weight of the edges matched. Constraint (2.2) ensures that a slot can display at most one advertisement. Constraint (2.3) limits the number of times each advertisement can be displayed.

This study focuses on the online version of the problem and proposes online algorithms to solve the problem. We assume that there is no knowledge of the arrival order of T in the adversarial order. For the sake of simplicity and tractability, the weight range for each ad i ($w_{ij} \in [L_i, U_i]$, $\forall j$ adjacent to i) is assumed to be known in advance. If the online algorithm finds a matching M for graph $\boldsymbol{G}$, then the objective value of the algorithm is $\sum_{(i,j) \in M} w_{ij}$. We use the notation of 'competitive ratio' to measure the performance of the online algorithm. The competitive ratio is defined as the ratio of the value obtained by the online algorithm (ALG) to the optimal offline objective value (OPT) given a bipartite graph $\boldsymbol{G}$. For every graph $\boldsymbol{G} = (A, T, E, w)$ and every order of T, the online algorithm is c-competitive if $ALG \geq c \cdot OPT$.

2.3 Deterministic Algorithms for Adversarial Order

In this section, we present deterministic algorithms for the display ads problem in the adversarial order. We have no knowledge of the arrival order of T over any bipartite graph $\boldsymbol{G} = (A, T, E, w)$. We assume that the range of the weights for ad $i \in A$ is $[L_i, U_i]$ and the capacity of it is C_i. A simple deterministic algorithm, called *Greedy*, is defined as follows:

Algorithm 2.1: Greedy

```
while a new node j ∈ T arrives do
    if all neighbors of j are unavailable (not or full connected) then
        continue;
    else
        match j to that available neighbor i which has the maximum value
          of w_ij;
    end
end
```

For Algorithm 2.1, we prove the following competitive ratio:

Theorem 2.1. *Algorithm 2.1 has a competitive ratio of* $\frac{1}{1+M_1}$ *for the display ads problem without free disposal* ($M_1 := \max(\frac{U_i}{L_i}), \forall i \in A$).

Proof. For simplicity, we assume that all capacities in A have the value of 1. Let $\mathbf{E}^*$ denote the set of optimal edges given by the offline algorithm, and let $\mathbf{E}'$ denote the set of edges produced by Algorithm 2.1. Let OPT and ALG be the objective values obtained by the offline algorithm and Algorithm 2.1, respectively.

The edges in $\mathbf{E}'$ can be divided into two types: $\mathbf{E}' \cap \mathbf{E}^*$ and $\mathbf{E}' \backslash \mathbf{E}^*$. The total value of the weights from $\mathbf{E}' \cap \mathbf{E}^*$ is the same in OPT and in ALG (let the value be K). For every edge $e_i \in \mathbf{E}' \backslash \mathbf{E}^*$, there may exist at most two edges f_i^1 and f_i^2 that are for $\mathbf{E}^* \backslash \mathbf{E}'$ as shown in Figure 2.1. (f_i^1(f_i^2): the edges incident with $b(a)$ in $\mathbf{E}^* \backslash \mathbf{E}'$, respectively). Let ALG_i denote the weight of the edge e_i and OPT_i denote the total weight obtained from the edges f_i^1 and f_i^2. We know that the weight of the edge f_i^1 is less than ALG_i, and the weight of the edge f_i^2 is less than or equal to U_i. So, we obtain $OPT_i < ALG_i + U_i$. It follows that $\frac{ALG_i}{OPT_i} > \frac{ALG_i}{ALG_i+U_i} \geq \frac{L_i}{L_i+U_i}$ because $ALG_i \geq L_i$.

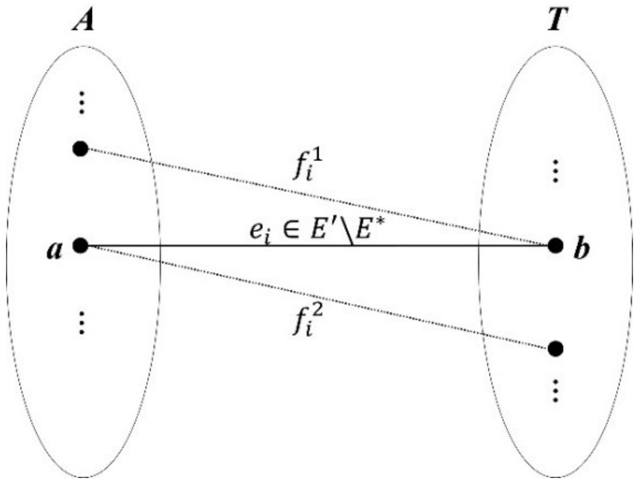

Figure 2.1: Example edge in $\mathbf{E}' \backslash \mathbf{E}^*$

Now, we analyze $\mathbf{E}^* \backslash \mathbf{E}'$. Let p denote $|\mathbf{E}' \backslash \mathbf{E}^*|$, noting that $\mathbf{E}^* \backslash \mathbf{E}' \subset \bigcup_{i=1,\ldots,p}(f_i^1 \cup f_i^2)$. Otherwise, the two nodes adjacent to the edge e ($e \in \mathbf{E}^* \backslash \mathbf{E}'$ but $e \notin \bigcup_{i=1,\ldots,p}(f_i^1 \cup f_i^2)$) have no degree in $\mathbf{E}'$. This finding contradicts the procedure used for Algorithm 2.1 because, in this case, the edge e would be included in $\mathbf{E}'$ while Algorithm 2.1

proceeds. Therefore, $ALG \geq K + \sum_{i=1,\dots,p} L_i$ and $OPT < K + \sum_{i=1,\dots,p}(L_i + U_i)$. According to these inequalities, we obtain

$$\begin{aligned}\frac{ALG}{OPT} &> \frac{K + \sum_{i=1,\dots,p} L_i}{K + \sum_{i=1,\dots,p}(L_i + U_i)} \geq \frac{\sum_{i=1,\dots,p} L_i}{\sum_{i=1,\dots,p}(L_i + U_i)} \\ &\geq \min(\frac{L_i}{L_i + U_i}) = \frac{1}{1 + \max(U_i/L_i)} = \frac{1}{1 + M_1}\end{aligned}$$

$\square$

For this problem, we consider a worst case in which the competitive ratio of any deterministic algorithms could be affected. For example, for ad $i \in A$, a deterministic algorithm has already matched i to C_i nodes in T. All the edges matched have low weights. Then, a node in T, which is the only neighbor of i, arrives. The weight of the edge between them is exceptionally high. To avoid this case, we propose Algorithm 2.2 which is based on the techniques developed by [118]. For Algorithm 2.2, we define variables, x_i, as the number of matched edges between ad i and nodes in T.

Algorithm 2.2: Greedy with sub_ads

```
for each i ∈ A do
    if L_i = U_i then
        k_i ← 1;
    else
        k_i ← min(C_i, ⌈ln (U_i/L_i)⌉)
    end
    Decompose a variable x_i into k_i variables x_{i0}, x_{i1}, ..., x_{i(k_i−1)} and set all
      variables to 0;
end
while a new node j ∈ T arrives do
    t ← 1;
    while t ≤ |A| do
        i ← a neighbor of j such that the weight of edge between them is the
          t^{th} highest among that of all edges adjacent to j;
        if i = ∅ then
            break;
        end
        Find an integer value p such that w_{ij} ∈ [L_i(U_i/L_i)^{p/k_i}, L_i(U_i/L_i)^{(p+1)/k_i});
        if x_{ip} < ⌊C_i/k_i⌋ then
            match j to i and x_{ip} ← x_{ip} + 1, break;
        else
            t ← t + 1;
        end
    end
end
```

For Algorithm 2.2, we prove the following competitive ratio as follows:

Theorem 2.2. *Algorithm 2.2 has a competitive ratio of* $\frac{1}{1+M_2}$ *for the display ads problem without free disposal* ($M_2 := \max\left(C_i \cdot \left\lfloor \frac{C_i}{k_i} \right\rfloor^{-1} \cdot (\frac{U_i}{L_i})^{\frac{1}{k_i}}\right)$ *such that* $k_i :=$ $\min(C_i, \left\lceil \ln \frac{U_i}{L_i} \right\rceil)$*,* $\forall i \in A$*).*

Proof. Let $\mathbf{E}^*$ denote the set of optimal edges given by the offline algorithm. Let

OPT and ALG be the objective values obtained by the offline and Algorithm 2.2, respectively. For Algorithm 2.2, each $i \in A$ is decomposed into k_i nodes (we call them *sub_ads*). Each sub_ad has one of the k_i disjoint ranges within $[L_i, U_i]$ and is matched to at most $\left\lfloor \frac{C_i}{k_i} \right\rfloor$ nodes in T. Let $\mathbf{E}''$ denote the set of edges produced by Algorithm 2.2 using the sub_ads.

We note that each edge in $\mathbf{E}^*$ can be mapped to one node (or sub_ad) matched in $\mathbf{E}''$. Let $e_{ij} \in \mathbf{E}^*$ and $w_{ij} \in [L_i(\frac{U_i}{L_i})^{\frac{p}{k_i}}, L_i(\frac{U_i}{L_i})^{\frac{p+1}{k_i}})$. There are two cases when $j \in T$ arrives: First, $x_{i,p}$ is less than $\left\lfloor \frac{C_i}{k_i} \right\rfloor$. In this case, the algorithm can match j to a sub_ad $s_1 \in A$ corresponding to $x_{u,v}$ (which may be $x_{i,p}$). We map e_{ij} to node $j \in T$. It follows that $w_{ij} \leq w_{s_1 j}$ because node j can be matched to at least the sub_ad corresponding to $x_{i,p}$. Second, $x_{i,p}$ is $\left\lfloor \frac{C_i}{k_i} \right\rfloor$. We map e_{ij} to sub_ad $s_2 \in A$ corresponding to $x_{i,p}$. Let $\mathbf{E}''(s_2) = \{e_{s_2 j} | e_{s_2 j} \in \mathbf{E}''\}$. In this instance, $\left\lfloor \frac{C_i}{k_i} \right\rfloor$ edges in $\mathbf{E}''(s_2)$ have already been matched and each edge has a weight of more than $L_i(\frac{U_i}{L_i})^{\frac{p}{k_i}}$. Let $ALG(\mathbf{E}''(s_2))$ be the total weight of the edges in $\mathbf{E}''(s_2)$. It follows that $ALG(\mathbf{E}''(s_2)) \geq \left\lfloor \frac{C_i}{k_i} \right\rfloor \cdot L_i(\frac{U_i}{L_i})^{\frac{p}{k_i}}$.

The edges in $\mathbf{E}^*$ can be divided into two types: $\mathbf{E}_1^*$ and $\mathbf{E}_2^*$. For $\mathbf{E}_1^*$, the edges are mapped from the first case and $\mathbf{E}_2^* = \mathbf{E}^* \backslash \mathbf{E}_1^*$. Let $OPT(\mathbf{E}_1^*)$ and $OPT(\mathbf{E}_2^*)$ be the total weight of the edges for $\mathbf{E}_1^*$ and $\mathbf{E}_2^*$, respectively. For the first case, we map $e_{ij} \in \mathbf{E}_1^*$ to node $j \in T$ and any two edges in $\mathbf{E}_1^*$ cannot be mapped to the same node in T. We have $w_{ij} \leq w_{s_1 j}$. Therefore, $OPT(\mathbf{E}_1^*) \leq ALG$. In the second case, we have $ALG(\mathbf{E}''(s_2)) \geq \left\lfloor \frac{C_i}{k_i} \right\rfloor \cdot L_i(\frac{U_i}{L_i})^{\frac{p}{k_i}}$. As $w_{ij} \leq L_i(\frac{U_i}{L_i})^{\frac{p+1}{k_i}}$, $ALG(\mathbf{E}''(s_2)) \geq \left\lfloor \frac{C_i}{k_i} \right\rfloor \cdot L_i(\frac{U_i}{L_i})^{\frac{p}{k_i}} \geq \left\lfloor \frac{C_i}{k_i} \right\rfloor \cdot (\frac{U_i}{L_i})^{-\frac{1}{k_i}} \cdot w_{ij}$. It follows that $w_{ij} \leq \left\lfloor \frac{C_i}{k_i} \right\rfloor^{-1} \cdot (\frac{U_i}{L_i})^{\frac{1}{k_i}} \cdot ALG(\mathbf{E}''(s_2))$. Note that each $i \in A$ matches at most C_i nodes in T. We have $\sum_{j \in T | e_{ij} \in \mathbf{E}_2^*} w_{ij} \leq C_i \cdot \left\lfloor \frac{C_i}{k_i} \right\rfloor^{-1} \cdot (\frac{U_i}{L_i})^{\frac{1}{k_i}} \cdot ALG(\mathbf{E}''(s_2)) = M_2 \cdot ALG(\mathbf{E}''(s_2))$. Because

$\sum_{s_2 \in A} ALG(\mathbf{E}''(s_2)) = ALG$, $\sum_{i \in A} \sum_{j \in T | e_{ij} \in \mathbf{E}_2^*} w_{ij} = OPT(\mathbf{E}^*(s_2)) \leq M_2 \cdot ALG$. We know $OPT = OPT(\mathbf{E}_1^*) + OPT(\mathbf{E}_2^*)$. Therefore, $OPT \leq (1 + M_2) \cdot ALG$ and the competitive ratio can be $\frac{1}{1+M_2}$. □

In this study, we propose an integrated algorithm that combines Algorithm 2.1 and Algorithm 2.2, and prove the following lemma:

Algorithm 2.3: Greedy + Greedy with sub_ads

if $M_1 \leq M_2$ **then**
| run Algorithm 2.1;
else
| run Algorithm 2.2;
end

$M_1 := \max(\frac{U_i}{L_i})$ and $M_2 := \max\left(C_i \cdot \left\lfloor \frac{C_i}{k_i} \right\rfloor^{-1} \cdot (\frac{U_i}{L_i})^{\frac{1}{k_i}}\right)$ such that $k_i := \min(C_i, \left\lceil \ln \frac{U_i}{L_i} \right\rceil), \forall i \in A$

Lemma 2.3. *Algorithm 2.3 has a competitive ratio of* $\max\left(\frac{1}{1+M_1}, \frac{1}{1+M_2}\right)$ *for the display ads problem without free disposal.*

Proof. The proofs for Theorems 2.1 and 2.2 prove Lemma 2.3. □

2.4 Upper Bounds of Deterministic Algorithms for Adversarial Order

In Section 2.3, we proved some competitive ratios of our deterministic algorithms for the display ads problem without free disposal. In this section, upper bounds on the competitive ratio of any deterministic algorithms are presented. We compare the gaps between the competitive ratios and the upper bounds depending on the size of values L_i, U_i, and C_i. [49] presented a simple upper bound for the problem.

Theorem 2.4. *No deterministic algorithm for the display ads problem achieves a competitive ratio better than $\frac{1}{2}$ even though free disposal is allowed.*

Proof. We consider an instance from $\boldsymbol{G} = (A, T, E, w)$ in which $A = \{i_1, i_2\}$ and $T = \{j_1, j_2\}$. The capacity of each advertisement is 1. When j_1 arrives first, we know that $w_{i_1 j_1}$ and $w_{i_2 j_1}$ have the same weight (let the weight be w). We arbitrarily match j_1 to i_1 (Fig. 2.2). Once j_1 has been matched, j_2 arrives. The edge incident to the same advertisement matched to j_1 is revealed only and the weight of the edge is w. As j_2 cannot be matched when it arrives, a deterministic algorithm can obtain w (Fig. 2.3). However, the optimal objective value is $2w$ (Fig. 2.4). Solid (dotted) lines mean matching (not matching) between an ad and a slot.

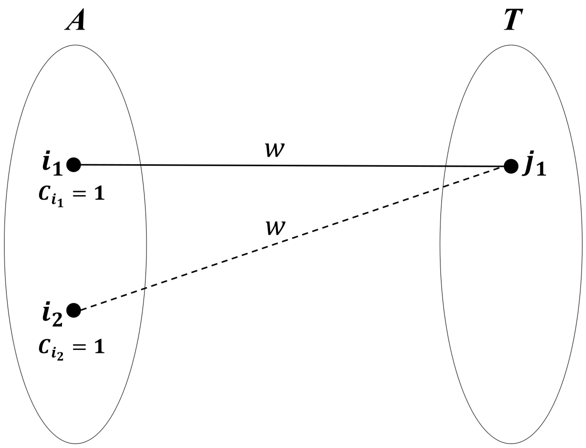

Figure 2.2: Result for deterministic algorithm when j_1 arrives

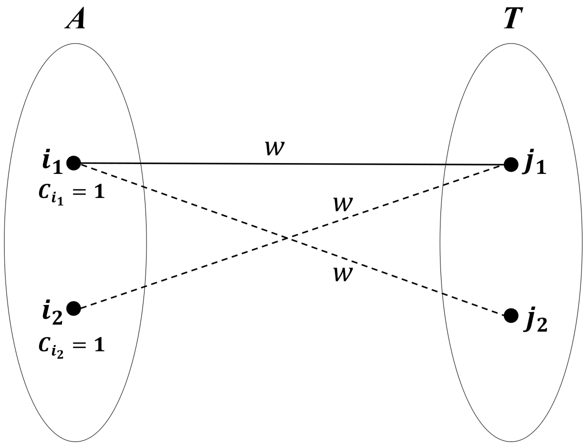

Figure 2.3: Result for deterministic algorithm when j_2 arrives

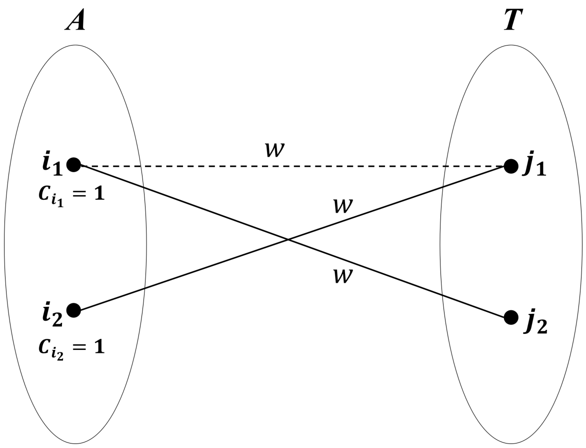

Figure 2.4: Optimal solution (offline version)

□

Using Theorem 2.4, we give the following lemma:

Lemma 2.5. *Algorithm 2.3 can be an optimal deterministic (online) algorithm when* $L_i = U_i,\ \forall i$.

Proof. When $L_i = U_i$ for every ad $i \in A$, the values of M_1 and M_2 in Algorithm 2.3 are all equal to 1. Hence, the competitive ratio of the algorithm can be $\frac{1}{2}$. As the upper bound of the competitive ratio is $\frac{1}{2}$ according to Theorem 2.4, Algorithm 2.3 can be an optimal deterministic (online) algorithm when $L_i = U_i,\ \forall i$. $\square$

To show that Algorithm 2.3 is effective even when $L_i \neq U_i$, we prove an upper bound on the competitive ratio for any deterministic algorithms and show that the gap between the upper bound and competitive ratio is not large. We present the following theorem, which is based on the techniques developed by [118], to get an upper bound.

Theorem 2.6. *No deterministic algorithm for the display ads problem without free disposal can have a competitive ratio larger than* $\min(\frac{1}{2}, K_1, K_2)$. $K_1 := \min\big(\frac{1}{(\frac{U_i}{L_i})^{\frac{1}{C_i}}}\big)$ *and* $K_2 := \min\big(\frac{1}{\lceil \ln \frac{U_i}{L_i} \rceil} \cdot \frac{e}{e-1}\big),\ \forall i \in A$.

Proof. We already proved $\frac{1}{2}$ in Theorem 2.4. First, we define a deterministic algorithm DA. Let OPT and ALG be the objective values obtained by the offline and deterministic algorithm DA, respectively. Next, we prove K_1 using an instance. Assume that $A = \{i\}$ and its capacity is C_i. A sequence $(j_1, j_2, \ldots, j_{(C_i+1)})$ of $C_i + 1$ nodes in T arrives online, and ad i and all nodes in T are adjacent. Let $w_{ij_k} = L_i(\frac{U_i}{L_i})^{\frac{k-1}{C_i}}$ for $1 \leq k \leq C_i + 1$. For a deterministic algorithm, we could find a value p such that j_p is not matched to i. When j_p is not matched to i in DA, the adversary stops the input sequence. Consider an instance with $T = \{j_1, j_2, \ldots, j_p\}$ arriving online. If $p = 1$, $ALG = 0$. It follows that the competitive ratio of this instance would be 0.

If $2 \le p \le C_i$, $ALG = \sum_{k=1}^{p-1} L_i (\frac{U_i}{L_i})^{\frac{k-1}{C_i}}$ and $OPT = \sum_{k=1}^{p} L_i (\frac{U_i}{L_i})^{\frac{k-1}{C_i}}$. The ratio is

$$\frac{ALG}{OPT} = \frac{\sum_{k=1}^{p-1} L_i (\frac{U_i}{L_i})^{\frac{k-1}{C_i}}}{\sum_{k=1}^{p} L_i (\frac{U_i}{L_i})^{\frac{k-1}{C_i}}} \le \frac{\sum_{k=1}^{p-1} L_i (\frac{U_i}{L_i})^{\frac{k-1}{C_i}}}{\sum_{k=2}^{p} L_i (\frac{U_i}{L_i})^{\frac{k-1}{C_i}}} = \frac{1}{(\frac{U_i}{L_i})^{\frac{1}{C_i}}} = K_1.$$

If $p = C_i + 1$, then

$$\frac{ALG}{OPT} = \frac{\sum_{k=1}^{p-1} L_i (\frac{U_i}{L_i})^{\frac{k-1}{C_i}}}{\sum_{k=2}^{p} L_i (\frac{U_i}{L_i})^{\frac{k-1}{C_i}}} = \frac{1}{(\frac{U_i}{L_i})^{\frac{1}{C_i}}} = K_1.$$

Therefore, an upper bound of the competitive ratio can be K_1.

We also show the proof to derive K_2. Assume that $A = \{i\}$ with capacity C_i. There are $\left\lceil \ln \frac{U_i}{L_i} \right\rceil$ types of nodes in T ($L_i \neq U_i$). Each type has C_i nodes that arrives continuously. A sequence $(j_1, j_1, \ldots, j_1, j_2, j_2, \ldots, j_2, \ldots, j_{\lceil \ln \frac{U_i}{L_i} \rceil}, j_{\lceil \ln \frac{U_i}{L_i} \rceil}, \ldots, j_{\lceil \ln \frac{U_i}{L_i} \rceil})$ of nodes in T arrives online, and ad i and all nodes in T are adjacent. Let $w_{ij_k} = L_i(e)^{k-1}$ for $1 \le k \le \left\lceil \ln \frac{U_i}{L_i} \right\rceil$. The adversary for any deterministic algorithm DA proceeds as follows: (Define $Y := \left\lceil \ln \frac{U_i}{L_i} \right\rceil$)

(1). C_i nodes corresponding to j_1 arrive online: If the number of nodes matched to i (say x_1) is less than or equal to C_i/Y, the adversary stops the input sequence. In other words, $x_1 \le C_1/Y$. The competitive ratio can be $\frac{ALG}{OPT} \le \frac{L_i \cdot (C_i/Y)}{L_i \cdot C_i} = \frac{1}{Y} \le K_2$. Otherwise, the adversary continues in (2).

(2). C_i nodes corresponding to j_2 arrive online: If the number of nodes matched to i (say $x_1 + x_2$) is less than or equal to $2 \cdot C_i/Y$, the adversary stops the input sequence. Because $x_1 > C_i/Y$ and $x_1 + x_2 \le 2 \cdot C_i/Y$, we have $x_2 \le C_i/Y$. Then ALG is at most $L_i \cdot (C_i/Y) + L_i \cdot e \cdot (C_i/Y)$. The competitive ratio can be $\frac{ALG}{OPT} \le \frac{L_i \cdot (e+1) \cdot (C_i/Y)}{L_i \cdot e \cdot C_i} = \frac{1}{Y} \cdot \frac{e+1}{e} \le K_2$. Otherwise, the adversary continues in (s), where s=3.

(s). ($3 \leq s \leq Y-1$) C_i nodes corresponding to j_s arrive online: If the number of nodes matched to i (say $\sum_{l=1}^{s} x_l$) is less than or equal to $s \cdot C_i/Y$, the adversary stops the input sequence. As $x_1 > C_i/Y$, $x_1 + x_2 > 2 \cdot C_i/Y$, ..., $x_1 + x_2 + \cdots + x_{s-1} > (s-1) \cdot C_i/Y$, and $x_1 + x_2 + \cdots + x_s \leq (s) \cdot C_i/Y$, we have $x_s \leq C_i/Y$. Then ALG is at most $L_i \cdot (C_i/Y) + L_i \cdot e \cdot (C_i/Y) + \cdots + L_i \cdot e^{s-1} \cdot (C_i/Y)$. The competitive ratio can be $\frac{ALG}{OPT} \leq \frac{L_i \cdot (e^{s-1} + \cdots + e + 1) \cdot (C_i/Y)}{L_i \cdot e^{s-1} \cdot C_i} = \frac{1}{Y} \cdot \frac{e^{s-1} + \cdots + e + 1}{e^{s-1}} \leq K_2$. Otherwise, the adversary continues in step (s+1).

(Y). C_i nodes corresponding to j_Y arrive online: By definition, $x_1 + x_2 + \cdots + x_Y \leq C_i = Y \cdot (C_i/Y)$. The competitive ratio can be $\frac{ALG}{OPT} \leq \frac{L_i \cdot (e^{Y-1} + \cdots + e + 1) \cdot (C_i/Y)}{L_i \cdot e^{Y-1} \cdot C_i} = \frac{1}{Y} \cdot \frac{e^{Y-1} + \cdots + e + 1}{e^{Y-1}} \leq K_2$. Therefore, an upper bound of the competitive ratio can be K_2. ($\because \lim_{n \to \infty} 1 + \frac{1}{e} + \cdots + \frac{1}{e^n} = \frac{e}{e-1}$).

$\square$

Table 2.1: Results for the gaps between competitive ratios and upper bounds

Case	(approaches)	**Ratio by Algorithm 2.3**	**Upper bound**
$U_i/L_i \simeq 1$	$\longrightarrow$	$1/2$	$1/2$
$C_i < U_i/L_i$	$\longrightarrow$	$\frac{1}{\left(1+C_i \cdot (\frac{U_i}{L_i})^{\frac{1}{C_i}}\right)}$	K_1
$C_i > U_i/L_i$	$\longrightarrow$	$\frac{1}{\left(1+\lceil \ln \frac{U_i}{L_i} \rceil\right)}$	K_2

Table 2.1 shows the results for the gaps between the competitive ratios and the upper bounds depending on the size of values C_i and the ratio of U_i to L_i (U_i/L_i). When the ratio of U_i to L_i is very small, the upper bound and lower bound approach

$\frac{1}{2}$. Otherwise, we divide the bounds into two cases according to the relative size of C_i compared to the ratio of U_i to L_i. When C_i is small, the upper bound approaches $K_1 := \min\left(\frac{1}{(\frac{U_i}{L_i})^{\frac{1}{C_i}}}\right)$ and the lower bound does $\frac{1}{\left(1+C_i\cdot(\frac{U_i}{L_i})^{\frac{1}{C_i}}\right)}$. When C_i is large, the upper bound approaches $K_2 := \min\left(\frac{1}{\lceil \ln \frac{U_i}{L_i}\rceil} \cdot \frac{e}{e-1}\right)$ and the lower bound does $\frac{1}{\left(1+\lceil \ln \frac{U_i}{L_i}\rceil\right)}$. The gaps between the upper and lower bounds are not large when the ratio of U_i to L_i is close to 1 or C_i is very large or small compared to the ratio of U_i to L_i. Algorithm 2.3 shows good performances for the analysis with the upper bound on the competitive ratio. In particular, Algorithm 2.3 might be a near-optimal algorithm, according to the value of C_i and $[L_i, U_i]$. When it comes to randomized algorithms, [117] proposed a near-optimal algorithm for the problem. For this reason, we do not cover randomized algorithms in this study.

2.5 Summary

This study dealt with the display ads problem, which is a generalization of the edge-weighted and capacitated online bipartite matching problem. Unlike the existing literature, this study presented the problem with the strict capacity constraint to reflect the realistic situation. To obtain bounded competitive ratios, we assumed that the online algorithms know the range of the weights $[L_i, U_i]$ for each ad i. Considering the strict capacity constraint, deterministic algorithms with worst-case guarantees were designed. We also proved upper bounds on the competitive ratio of any deterministic algorithms. We derived that the deterministic algorithm may be a near-optimal algorithm according to the capacity and weight range. The results not only showed the improved worst-case guarantees but also led to theoretical contributions to research on the edge-weighted and capacitated online bipartite matching problem.

Chapter 3

Online Advertising Assignment Problem in Probabilistic Order

3.1 Problem Description and Literature Review

As mentioned in Chapter 2, an online advertising assignment problem can correspond to a bipartite matching problem in graph theory. Unlike the matching problem in which cardinality is maximized, the objective of the online advertising assignment problem is to find connections, through which revenue is maximized, between the advertisements and the slots. The connections between the advertisements and the slots are generally decided either by auction or through contracts [99]. Each edge (connection) has a weight. The weight of the edge might be a prediction of click-through probability, an estimate of targeting quality, or a bid submitted by the advertiser [16, 49]. The weight corresponding to the edge is realized when an advertisement is assigned to a slot.

This study focuses on the online version of the online advertising assignment problem, which is similar to Chapter 2. In reality, information on weights is not known beforehand, making the problem uncertain. [46, 84, 109]. In other words, we define a bipartite graph for which information about the nodes on the left-hand side is known in advance, and the nodes on the right-hand side arrive online (one node at a time). The nodes on the left-hand side represent advertisements and those on the right-hand side represent slots. When a node on the right-hand side arrives, the edges and weights incident of the node are revealed. An online algorithm of the problem selects one of the edges (an advertisement is displayed on the slot) or discards them (no advertisement is displayed on the slot). The decision is irrevocable [2, 17, 24, 26, 29, 50, 57, 61, 65, 71, 77, 78, 81, 94, 96, 105, 119].

This chapter covers the display ads problem as Chapter 2 covered. The problem is a generalization of the edge-weighted and capacitated online bipartite matching problem. The property referred to as *free disposal*, which was introduced by [49], is also excluded in this chapter. The definition of the free disposal assumption is that each node on the left-hand side is allowed to be matched more times than its capacity, (c), but the publishers gain only for the 'c' highest weights matched. In other words, the assumption allows for violating the capacity constraint. If some advertisers are sensitive to the number of times their advertisements are displayed, the solutions with the assumption might cause issues with trust. That is because there is a possibility that an advertisement can be displayed more times than its capacity, while other advertisements miss the chance to be displayed. Also, the display ads problem that allows for the free disposal assumption is challenging to apply to other types of problems (e.g., scheduling or resource allocation) in which resources, such as humans and machines, are strictly limited. Like Chapter 2, this chapter deals with 'the display ads problem without free disposal.'

This study presents the analyses of probabilistic orders to the problem. The adversarial order covered in Chapter 2 focuses on the worst-case analysis which considers all possible cases. That is, given an uncertainty set, the adversarial order means a non-parametric approach. However, in real problems, companies can stochastically estimate the input sequence by using historical data. It means that we make certain assumptions about a given uncertainty set through historical data. It is natural to consider probabilistic orders for the display ads problem, which can be expressed as a parametric approach [48]. In Chapter 3, we cover the display ads problem in probabilistic orders.

There has been some literature on the problem in the probabilistic orders. [86] developed a $\frac{1}{8}$-approximation algorithm for the online weighted bipartite matching problem in the random order. [3] and [65] proved a competitive ratio of $1 - \frac{1}{e}$ and 0.6534 for the online vertex-weighted bipartite matching problem in the random order, respectively. [83] presented an algorithm that is a generalization of the secretary problem to solve the edge-weighted bipartite online matching problem in the random order.

Two probabilistic order models (*known IID* and *random permutation*) are considered in this study. The two probabilistic orders are generally used as stochastic input models of the online matching problem [99]. This study presents stochastic online algorithms with scenario-based stochastic programming and Benders decomposition. The stochastic online algorithm is based on the algorithms presented in [49] and [90]. Upon each arrival of a node on the right-hand side, the algorithm estimates future scenarios of remaining slots and solves optimization problems to make a competitive decision for the current slot. The algorithm is used in this study to handle the uncertainty reasonably. Numerical experiments on various future scenarios are conducted to show better performances over the primal-dual algorithm of [49]. Chapter 2 is a theoretical perspective, while Chapter 3 is a practical perspective.

The rest of this section is organized as follows. In Section 3.2, we define the display ads problem without free disposal in the probabilistic order. Section 3.3 introduces stochastic online algorithms with scenario-based stochastic programming and Benders decomposition for the probabilistic orders. Section 3.4 provides numerical results of the stochastic online algorithm which is presented in Section 3.3. Section 3.5 offers our contributions and conclusions of this study.

3.2 Display Ads Problem in Probabilistic Order

For an edge-weighted bipartite graph $\boldsymbol{G} = (A, T, E, w)$, a mathematical formulation for the display ads problem is as follows:

$$\max \sum_{i=1}^{|A|} \sum_{j=1}^{|T|} w_{ij} x_{ij} \tag{3.1}$$

$$\text{s.t.} \sum_{i=1}^{|A|} x_{ij} \leq 1 \ \forall j \in T \tag{3.2}$$

$$\sum_{j=1}^{|T|} x_{ij} \leq C_i \ \forall i \in A \tag{3.3}$$

$$x_{ij} \in \{0, 1\} \ \forall i \in A, j \in T \tag{3.4}$$

The notations used in the formulation are the same as those in Sections 2.2. The binary decision variable, x_{ij}, is 1 if ad $i \in A$ is matched to slot $j \in T$; 0 otherwise. The objective function (3.1) maximizes the total weight of the edges matched. Constraint (3.2) ensures that a slot can display at most one advertisement. Constraint (3.3) limits the number of times each advertisement can be displayed.

This study focuses on the online version of the problem and proposes online algorithms to solve the problem. In online environment, we do not know any information of T, E, and w, except $|T|$. The set of T arrives online, one node at a time. When a node $j \in T$ arrives, all edges incident to j as well as the weights, w_{ij}, of each are revealed. The algorithm matches a connection between a node j and one of the advertisements available or leaves the node unmatched. The decision made is irrevocable. Unlike the adversarial order in Chapter 2, this chapter deals with probabilistic orders. There is an arrival order of T under the probabilistic structure in the probabilistic order. Two probabilistic order models (*known IID* and *random permutation*) are considered. The known IID model starts with the information on types of nodes in T. We know a probability distribution on a collection of node types. The random permutation model assumes that the nodes in T arrive in a uniform random permutation. The details of the probabilistic orders are presented in Section 3.3.

3.3 Stochastic Online Algorithms for Probabilistic Order

In this section, we assume that publishers already know information on the probability distribution for the future coming slots. Stochastic online algorithms are proposed to solve the display ads problem in the probabilistic order. The algorithm combines the primal-dual algorithms developed by [49] with scenario-based stochastic programming and Benders decomposition proposed by [90]. We use two-stage stochastic programming to handle the stochastic programming problem. A brief explanation of the two-stage stochastic is in 3.3.1.

The algorithm in the probabilistic orders is a practical approach. We conduct numerical experiments of the stochastic online algorithm, and the results are presented in Section 3.4. We uses two stochastic models according to information of the sequence on the right-hand nodes: the *known IID* and *random permutation* models.

3.3.1 Two-Stage Stochastic Programming

Stochastic programming is an approach for modeling optimization problems that include uncertainty [115]. In general, we cover optimization problems on condition that all input parameters (information) are given. We call them deterministic optimization problems. The optimization problem can be expressed as follows:

$$\max_{x \in X} f(x, \xi)$$

where the objective function is $f(\cdot, \cdot) : X \times \Xi \to \mathbb{R}$. $x \in X$ is a vector of decision variables and $\xi \in \Xi$ is a parameter.

However, in practice, optimization problems may be faced with the situation in which there are some unknown parameters and uncertain situations as mentioned in Section 1.1.1. If we know and estimate the probability distribution of the parameter (ξ), stochastic programming can be applied to solve the problem. The objective of the stochastic programming model is to find some solution that not only is feasible but also performs well on average. That is, in the objective function, there is an expectation term that includes a function consisting of decision variables and random variables. We focus on maximizing (or minimizing) the objective function including the expectation term. The basic stochastic programming model is as follows:

$$\max_{x \in X} \mathbb{E}[f(x,\xi)]$$

where ξ is a random variable with a known probability distribution.

One of the most widely applied and studied stochastic programming models is two-stage stochastic programming. A general two-stage (linear) stochastic programming problem can be expressed as follows:

$$\max_{x \in X} \{g(x) := c^T x + \mathbb{E}[Q(x,\xi)]\}$$

where $Q(x,\xi)$ is the optimal value of the second-stage problem

$$\begin{aligned} Q(x,\xi) := \max_{y \in \mathbb{R}^m} \ & q(\xi)^T y \\ \text{s.t. } & y \in \mathcal{Y}(x,\xi) \end{aligned}$$

where x is the vector of the first-stage decision variables, y is the second-stage decision variable vector. $\mathcal{Y}(x,\xi)$ represents the set of feasible solutions for a given x. In the two-stage stochastic programming, we solve the problem in the first stage before the uncertainty is revealed. The second-stage problem can be represented as an optimization problem that finds an optimal solution, which satisfies the constraint, after the uncertainty is revealed.

To solve the problem numerically, we assume that $|\Xi|$ is finite and the probability of each realization can be obtained. For example, there are K scenarios (realizations) of the uncertain parameter, say $\xi_1, \ldots, \xi_K$, and the probability of scenario ξ_k is p_k

($\sum_{k=1}^{K} p_k = 1$). Hence, we can express the expectation term of the objective function as follows:

$$\mathbb{E}[Q(x, \xi)] = \sum_{k=1}^{K} p_k Q(x, \xi_k)$$

The two-stage stochastic programming problem can be formulated as a deterministic optimization problem as follows:

$$\begin{aligned}
\max\ & c^T x + \sum_{k=1}^{K} p_k q(\xi_k)^T y_k \\
\text{s.t. } & x \in X \\
& y_k \in \mathcal{Y}(x, \xi_k)\ \forall k = 1, 2, \ldots, K
\end{aligned}$$

3.3.2 Known IID model

For the known IID model, we assume that there are some types of nodes in T. For a collection K of node types, the publishers know a probability distribution on K in advance. For each slot, a node type $k \in K$ is drawn from the probability distribution. We suppose that the j^{th} slot has just arrived (such that k_j, which is a node type at slot j, is revealed) and that the publishers decide an advertisement to be displayed on the slot. Parameters and decision variables for a stochastic programming formulation are introduced as follows:

Table 3.1: Parameters and decision variables

Ω_j	Set of future sample scenarios at slot j
p^ω	Probability of scenario ω ($\omega \in \Omega_j$)
w_{ik}	Weight of the edge between ad i and type k
T^ω_{jk}	Number of slots for type k in scenario ω
C_i^{left}	Capacity left for ad i
x_i	Binary decision variable, whose value is 1 if ad i is allocated to slot j, 0 otherwise
y^ω_{ik}	Number of slots for type k allocated to ad i for scenario ω

A stochastic programming formulation at the time of the j^{th} slot is as follows:

$$\max \sum_i w_{ik_j} x_i + \sum_{\omega\in\Omega_j} p^\omega \sum_i \sum_k w_{ik} y^\omega_{ik} \tag{3.5}$$

$$\text{s.t.} \sum_{i=1}^{|A|} x_i \leq 1 \tag{3.6}$$

$$\sum_{i=1}^{|A|} y^\omega_{ik} \leq T^\omega_{jk} \quad \forall \omega \in \Omega_j, k \in K \tag{3.7}$$

$$x_i + \sum_k y^\omega_{ik} \leq C_i^{left} \quad \forall \omega \in \Omega_j, i \in A \tag{3.8}$$

$$\boldsymbol{x} \in \mathbb{B}^{|A|} \tag{3.9}$$

$$y^\omega_{ik} \in \mathbb{N} \quad \forall i \in A, k \in K, \omega \in \Omega_j \tag{3.10}$$

The objective function (3.5) maximizes the weight for the j^{th} slot and the expected total weight obtained from the remaining future slots. Constraint (3.6) ensures that the j^{th} slot can display at most one advertisement. Constraint (3.7) guarantees that the number of slots for type k allocated to all advertisements in scenario ω cannot exceed T_{jk}^{ω}. Constraint (3.8) limits the capacity left for each advertisement and each scenario. Constraints (3.9) and (3.10) define x_i and y_{ik}^{ω} as binary and integer variables, respectively.

The stochastic formulation can be decomposed using Benders decomposition [12]:

Master problem

$$\max \sum_{i} w_{ik_j} x_i + \sum_{\omega \in \Omega_j} p^{\omega} S(\boldsymbol{x}, \omega) \tag{3.11}$$

$$\text{s.t.} \sum_{i=1}^{|A|} x_i \leq 1 \tag{3.12}$$

$$\boldsymbol{x} \in \mathbb{B}^{|A|} \tag{3.13}$$

Slave problem (for each $\boldsymbol{x}$ and ω)

$$S(\boldsymbol{x}, \omega) = \max \sum_{i} \sum_{k} w_{ik} y_{ik}^{\omega} \tag{3.14}$$

$$\text{s.t.} \sum_{i=1}^{|A|} y_{ik}^{\omega} \leq T_{jk}^{\omega} \quad \forall k \in K \tag{3.15}$$

$$\sum_{k} y_{ik}^{\omega} \leq C_i^{left} - x_i \quad \forall i \in A \tag{3.16}$$

$$y_{ik}^{\omega} \in \mathbb{N} \quad \forall i \in A, k \in K \tag{3.17}$$

For each $\boldsymbol{x}$ and ω, we should obtain the objective value of the slave problem. The value can be approximated by using the dual slave problem. α_k^{ω} and β_i^{ω} are dual variables corresponding to the first and second types of constraint for each slave problem, respectively. Using the weak duality theorem, we show that $S(\boldsymbol{x}, \omega) \leq \sum_k T_{jk}^{\omega}\alpha_k^{\omega} + \sum_i (C_i^{left} - x_i)\beta_i^{\omega}$ for every $\boldsymbol{x}$ and ω. The objective value of the dual problem can be a cut for the master problem:

<u>Dual slave problem (for each $\boldsymbol{x}$ and ω)</u>

$$\min \sum_k T_{jk}^{\omega}\alpha_k^{\omega} + \sum_i (C_i^{left} - x_i)\beta_i^{\omega} \tag{3.18}$$

$$\text{s.t. } \alpha_k^{\omega} + \beta_i^{\omega} \geq w_{ik} \quad \forall i \in A, k \in K \tag{3.19}$$

$$\alpha_k^{\omega} \geq 0, \beta_i^{\omega} \geq 0, \quad \forall i \in A, k \in K \tag{3.20}$$

<u>Master problem with added cuts</u>

$$\max \sum_i w_{ik_j} x_i + \sum_{\omega \in \Omega_j} p^{\omega} S^{\omega} \tag{3.21}$$

$$\text{s.t.} \sum_{i=1}^{|A|} x_i \leq 1 \tag{3.22}$$

$$\begin{aligned} S^{\omega} \leq & \sum_k T_{jk}^{\omega}\alpha_k^{\omega} \\ & + \sum_i (C_i^{left} - x_i)\beta_i^{\omega} \quad \forall \omega \in \Omega_j \end{aligned} \tag{3.23}$$

$$\boldsymbol{x} \in \mathbb{B}^{|A|} \tag{3.24}$$

In terms of optimal solutions, the objective function of the master problem can be replaced by $\sum_i w_{ik_j} x_i + \sum_{\omega \in \Omega_j} p^{\omega} \cdot [\sum_k T_{jk}^{\omega}\alpha_k^{\omega} + \sum_i (C_i^{left} - x_i)\beta_i^{\omega}]$. If the constant terms are eliminated, then the objective function can be $\sum_i w_{ik_j} x_i - \sum_{\omega \in \Omega_j} p^{\omega} \cdot$

$(\sum_i \beta_i^\omega x_i)$. By using the values of the dual variables β_i^ω, we develop a stochastic online algorithm with the primal-dual algorithm, which is presented in Section 3.3.4.

3.3.3 Random permutation model

For the random permutation model, we assume that there are $|T|$ node types in T and the number of slots for each type is 1. For a collection K of node types, it becomes $K = |T|$ and $T = \{1, 2, \cdots, |T|\}$. That is, a sequence on the right-hand side nodes becomes one of the random permutations of T. The probability of each sequence is identical. Like with the known IID model, we suppose that the j^{th} slot has just arrived (such that a type at slot j is revealed) and that the publishers choose an advertisement to be displayed on the slot. Parameters and decision variables for a stochastic programming formulation are introduced as follows:

Table 3.2: Parameters and decision variables

Ω_j	Set of future sample scenarios at slot j, $\|\Omega_j\| = (\|T\| - j)!$
p_j	Probability of each scenario at slot j, $p_j = \frac{1}{\|\Omega_j\|}$
w_{ik}^ω	Weight of the edge between ad i and slot k in scenario ω ($k \geq j+1$)
C_i^{left}	Capacity left for ad i
x_i	Binary decision variable, whose value is 1 if ad i is allocated to slot j, 0 otherwise
x_{ik}^ω	Binary decision variable, whose value is 1 if ad i is allocated to slot k in scenario ω; 0 otherwise ($k \geq j+1$)

A stochastic programming formulation at the time of the j^{th} slot is as follows:

$$\max \sum_{i} w_{ij} x_i + \sum_{\omega \in \Omega_j} p_j \sum_{i} \sum_{k \geq j+1} w_{ik}^{\omega} x_{ik}^{\omega} \tag{3.25}$$

$$\text{s.t.} \sum_{i=1}^{|A|} x_i \leq 1 \tag{3.26}$$

$$\sum_{i=1}^{|A|} x_{ik}^{\omega} \leq 1 \quad \forall \omega \in \Omega_j, k \geq j+1 \tag{3.27}$$

$$x_i + \sum_{k \geq j+1} x_{ik}^{\omega} \leq C_i^{left} \quad \forall \omega \in \Omega_j, i \in A \tag{3.28}$$

$$\boldsymbol{x} \in \mathbb{B}^{|A|} \tag{3.29}$$

$$x_{ik}^{\omega} \in \mathbb{B} \quad \forall i \in A, k \geq j+1, \omega \in \Omega_j \tag{3.30}$$

The objective function (3.25) maximizes the weight for the j^{th} slot and the expected total weight obtained by the remaining future slots. Constraints (3.26) and (3.27) ensure that each slot of each scenario can display at most one advertisement. Constraint (3.28) limits the capacity left for each advertisement and each scenario. Constraints (3.29) and (3.30) define x_i and x_{ik}^{ω} as binary variables, respectively. Like the known IID model, the stochastic formulation can be decomposed by using Benders decomposition [12]:

Master problem

$$\max \sum_i w_{ij}x_i + \sum_{\omega \in \Omega_j} p_j S(\boldsymbol{x}, \omega) \tag{3.31}$$

$$\text{s.t.} \sum_{i=1}^{|A|} x_i \leq 1 \tag{3.32}$$

$$\boldsymbol{x} \in \mathbb{B}^{|A|} \tag{3.33}$$

Slave problems (for each $\boldsymbol{x}$ and ω)

$$S(\boldsymbol{x}, \omega) = \max \sum_i \sum_k w_{ik}^{\omega} x_{ik}^{\omega} \tag{3.34}$$

$$\text{s.t.} \sum_{i=1}^{|A|} x_{ik}^{\omega} \leq 1 \quad \forall k \geq j+1 \tag{3.35}$$

$$\sum_{k \geq j+1} x_{ik}^{\omega} \leq C_i^{left} - x_i \quad \forall i \in A \tag{3.36}$$

$$x_{ik}^{\omega} \in \mathbb{B} \quad \forall i \in A, k \geq j+1 \tag{3.37}$$

For each $\boldsymbol{x}$ and ω, we should obtain the objective value of the slave problem. The value can be approximated by using the dual slave problem. α_k^{ω} and β_i^{ω} are dual variables corresponding to the first and second types of constraint for each slave problem, respectively. Using the weak duality theorem, we show that $S(\boldsymbol{x}, \omega) \leq \sum_k T_{jk}^{\omega} \alpha_k^{\omega} + \sum_i (C_i^{left} - x_i)\beta_i^{\omega}$ for every $\boldsymbol{x}$ and ω. The objective value of the dual problem can be a cut for the master problem:

Dual slave problem (for each $\boldsymbol{x}$ and ω)

$$\min \sum_k \alpha_k^\omega + \sum_i (C_i^{left} - x_i)\beta_i^\omega \tag{3.38}$$

$$\text{s.t. } \alpha_k^\omega + \beta_i^\omega \geq w_{ik} \quad \forall i \in A, k \geq j+1 \tag{3.39}$$

$$\alpha_k^\omega \geq 0, \beta_i^\omega \geq 0, \quad \forall i \in A, k \geq j+1 \tag{3.40}$$

Master problem with added cuts

$$\max \sum_i w_{ij} x_i + \sum_{\omega \in \Omega_j} p_j S^\omega \tag{3.41}$$

$$\text{s.t.} \sum_{i=1}^{|A|} x_i \leq 1 \tag{3.42}$$

$$S^\omega \leq \sum_{k \geq j+1} \alpha_k^\omega + \sum_i (C_i^{left} - x_i)\beta_i^\omega \quad \forall \omega \in \Omega_j \tag{3.43}$$

$$\boldsymbol{x} \in \mathbb{B}^{|A|} \tag{3.44}$$

In terms of optimal solutions, the objective function of the master problem can be replaced by $\sum_i w_{ij} x_i + \sum_{\omega \in \Omega_j} p_j \cdot [\sum_{k \geq j+1} \alpha_k^\omega + \sum_i (C_i^{left} - x_i)\beta_i^\omega]$. If the constant terms are eliminated, then the objective function can be $\sum_i w_{ij} x_i - \sum_{\omega \in \Omega_j} p_j \cdot (\sum_i \beta_i^\omega x_i)$. By using the values of the dual variables β_i^ω, we develop a stochastic online algorithm with the primal-dual algorithm, which is presented in Section 3.3.4.

3.3.4 Stochastic approach using primal-dual algorithm

[49] provided a primal-dual algorithm to obtain a good competitive ratio for the display ads problem. Primal and dual linear programming (LP) formulations for the display ads problem are as follows:

Primal LP

$$\max \sum_{i=1}^{|A|} \sum_{j=1}^{|T|} w_{ij} x_{ij} \tag{3.45}$$

$$\text{s.t.} \sum_{i=1}^{|A|} x_{ij} \leq 1 \quad \forall j \in T \tag{3.46}$$

$$\sum_{j=1}^{|T|} x_{ij} \leq C_i \quad \forall i \in A \tag{3.47}$$

$$x_{ij} \geq 0 \quad \forall i \in A, j \in T \tag{3.48}$$

Dual LP

$$\min \sum_{j=1}^{|T|} \alpha_j + \sum_{i=1}^{|A|} \beta_i \tag{3.49}$$

$$\text{s.t. } \alpha_j + \beta_i \geq w_{ij} \quad \forall i \in A, j \in T \tag{3.50}$$

$$\alpha_j \geq 0, \;\; \beta_j \geq 0 \; \forall i \in A, j \in T \tag{3.51}$$

The algorithm uses the dual variables β_i in dual LP to display an advertisement on a slot. First, the dual variables β_i are all initialized to 0. When a slot $j \in T$ arrives online, we select an advertisement that maximizes $w_{ij} - \beta_i$ among the advertisements available, and the selected advertisement is displayed on slot j. If $w_{ij} - \beta_i < 0$, then leave slot j unassigned because the solution is infeasible for the dual. If the advertisement is displayed, then set $x_{ij} := 1$, $\alpha_j := w_{ij} - \beta_i$, and β_i is updated with one of the update rules (i.e. greedy, uniform weighting, or exponential weighting). The rules were proposed by [49] as a means to obtain theoretical competitive ratios. The algorithm proceeds until all slots in T arrive. At each iteration, the primal solution gives a feasible integer solution, and the dual solution is also feasible. The value of β_i plays a role in adjusting the weight w_{ij} by increasing β_i as the number of ad i displayed increases. Therefore, it is important for the publishers to decide an appropriate value for β_i and use it in the algorithm.

This study presents a stochastic online algorithm that is based on the primal-dual algorithm. The algorithm updates β_i with $\sum_{\omega \in \Omega_j} p^\omega \cdot \beta_i^\omega$ (in the known IID model) or $\sum_{\omega \in \Omega_j} p_j \cdot \beta_i^\omega$ (in the random permutation model), which is obtained by the stochastic programming formulation. Compared with β_i, these values are more likely to be appropriate because they reflect stochastic information. A primal-dual algorithm with stochastic information is shown as follows:

Algorithm 3.1: Primal-dual algorithm with stochastic information

$x_{ij} \leftarrow 0 \;\; \forall i \in A, j \in T$;

$\beta_i \leftarrow 0,\; C_i^{left} \leftarrow C_i \;\; \forall i \in A$;

$t \leftarrow 1$;

while *a new node* $j \in T$ *arrives* **do**

  Select $i \in A$ which maximizes the value $w_{ij} - \beta_i$ and satisfies $C_i^{left} > 0$;

  if $i \in A$ *is selected* **then**

    $x_{ij} \leftarrow 1$ and $C_i^{left} \leftarrow C_i^{left} - 1$;

    update β_i by one rule (e.g., greedy, uniform weighting, or exponential weighting);

  end

  if $t \equiv 0 \; (mod \; \Delta)$ **then**

    solve the stochastic programming formulation at the time of t^{th} slot to obtain β_i^{ω};

    update either $\beta_i \leftarrow \lambda \cdot \beta_i + (1-\lambda) \cdot \sum_{\omega \in \Omega_j} p^{\omega} \cdot \beta_i^{\omega}$ (known IID) $\forall i \in A$ or $\beta_i \leftarrow \lambda \cdot \beta_i + (1-\lambda) \cdot \sum_{\omega \in \Omega_j} p_j \cdot \beta_i^{\omega}$ (random permutation), $\forall i \in A$ $(0 \leq \lambda \leq 1)$;

  end

  $t \leftarrow t + 1$;

end

This algorithm selects ad $i \in A$ by using updated values β_i. If the stochastic programming formulation is solved each time a slot arrives, then the algorithm takes much computation time. To shorten the computation time, the algorithm uses the stochastic technique only for every Δ slots. A parameter λ $(0 \leq \lambda < 1)$ is introduced to adjust the effect of the stochastic technique. The effect is higher when the value of λ is small. Section 3.4 provides the numerical results obtained with the stochastic online algorithm.

3.4 Computational Experiments

In this section, we analyze the primal-dual algorithm with stochastic information (Algorithm 3.1) through numerical experiments. The algorithm was run with JAVA language in Windows 7 on a PC with an Intel(R) Core(TM) i5-4690 CPU 3.5GHz with 16.00 GB of RAM. IBM ILOG CPLEX version 12.8 was used to obtain the dual variables β_i^w for each scenario. For these experiments, we use the 'experimental ratio' as the ratio of the value obtained by Algorithm 3.1 to the optimal offline objective value at each bipartite graph. We used instances in this experiment: 50 advertisements ($|A| = 50$) and 200 slots ($|T| = 200$). The capacity of each advertisement was set to 4. The ratio of $\frac{U_i}{L_i}$ was limited to 10 for each advertisement. The number of node types was set to 20 ($|K| = 20$), and the sequence of slots assumed to follow a multinomial distribution in the known IID model.

To obtain values for β_i^ω, we solve the (dual) slave problem for each scenario. Many future scenarios are drawn for the time of each slot. If we consider all the scenarios drawn, the computation time might be time-consuming for the (dual) slave problems. Figures 3.1 and 3.2 present the experimental ratios and computation times for different values of $|\Omega_j|$. We used $\lambda = 0$ and $\Delta = 1$. The values in Figures 3.1 and 3.2 mean the average values for 100 data sets.

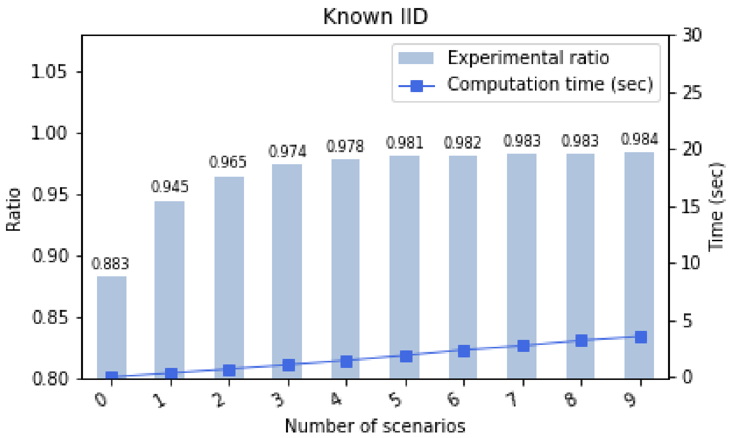

Figure 3.1: Results for different value of $|\Omega_j|$ (Known IID)

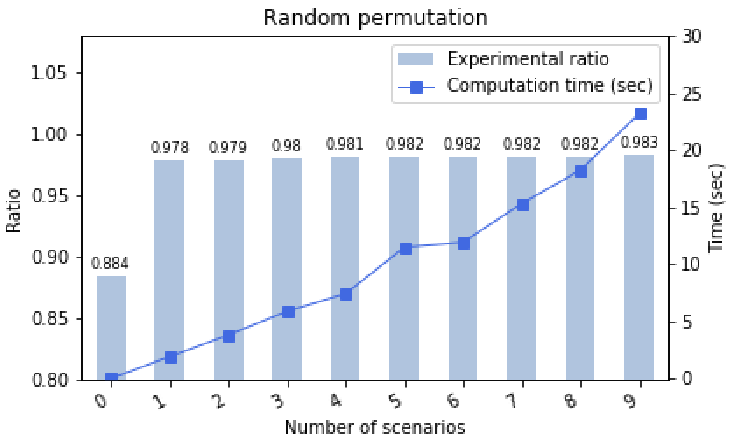

Figure 3.2: Results for different value of $|\Omega_j|$ (Random permutation)

The experimental ratio increased by 6.2% (known IID) and 9.4% (random permutation) when we apply Algorithm 3.1 ($|\Omega_j| = 1$). The computation times increased linearly in both of the two probabilistic models as $|\Omega_j|$ increases, but the increasing rate was higher in the random permutation model. The experimental ratio tended to increase with increasing values of $|\Omega_j|$. The increase in the experimental ratio was shown to be more apparent in the known IID model than in the random permutation model. The known IID model showed statistically significant differences between the average experimental ratios from $|\Omega_j| = 0$ to 5. Meanwhile, the random permutation showed statistically significant differences from $|\Omega_j| = 0$ to 4. This implies that the adjusted dual variable β_i^ω calculated by using future scenarios showed better performance than β_i obtained by the previous update rule. However, it does not always mean that the performance improves as the number of future scenarios increases. Hence, considering a trade-off between the experimental ratios and computation times presented in Figures 3.1 and 3.2, we decided to set $|\Omega_j|$ to 5 (in the known IID model) and 4 (in the random permutation model) for the following analysis.

Experimental results may vary in different probability distributions. However, in general, even when a greedy-type algorithm to solve an online problem is applied, the experimental ratios are much better than proved competitive ratios in most cases [110]. Even without generating a scenario, the experimental ratio was higher than 88% in both of the two probabilistic orders. In fact, the competitive ratio is greatly affected because of a minority of the worst cases. Conducting Algorithm 3.1 with an arbitrary scenario is similar to conducting a randomized algorithm. If a randomized algorithm is applied, results from the worst cases might not be more likely to occur [90, 79]. Moreover, the two probabilistic orders used in this model have relatively simple structures, so the deviation of the expectation values depending on scenarios might not be large. It might be difficult to improve the performance with a small number of scenarios when a problem has many worst cases or complex order sequence. It is crucial that we decide an appropriate value of $|\Omega_j|$ depending on problems in terms of both ratios and computation times.

Sensitivity analysis of the parameters λ and Δ was performed for the two stochastic models to derive meaningful insights about Algorithm 3.1. Figures 3.3 - 3.6 present the experimental ratios for different values of the parameters λ and Δ. The values in Figures 3.3 - 3.6 are average values for 100 data sets. Figures 3.3 - 3.6 show that Algorithm 3.1 performed better as the value of Δ decreased for the two probabilistic orders. It means that the more frequently we use the adjusted dual variable β_i^{ω}, the more likely we obtain high experimental ratios. The differences in the experimental ratio were as much as 6.2%, depending on Δ (1 to 50). In addition, the figures show that the experimental ratios were affected by the value of λ. The effect of the value of λ tended to be no higher than that of the value of Δ. Overall, $\lambda = 0.3$ showed the best results for these experiments.

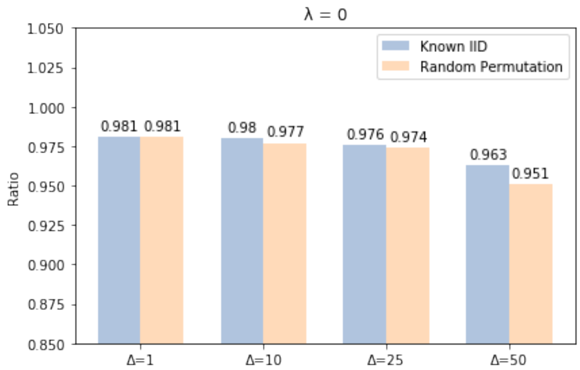

Figure 3.3: Experimental ratios for different values of Δ ($\lambda = 0$)

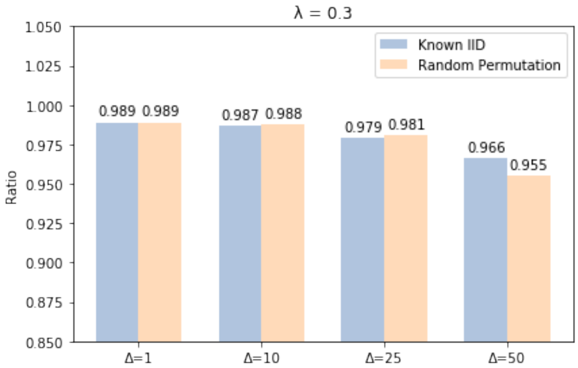

Figure 3.4: Experimental ratios for different values of Δ ($\lambda = 0.3$)

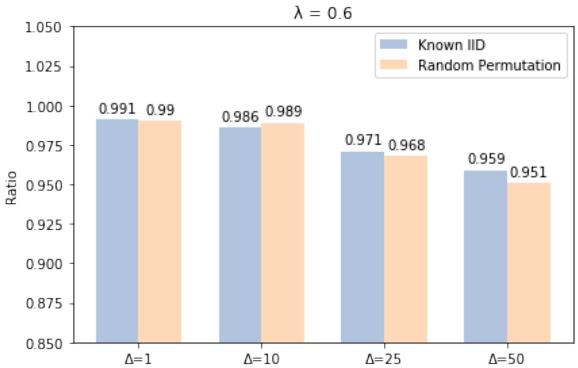

Figure 3.5: Experimental ratios for different values of Δ ($\lambda = 0.6$)

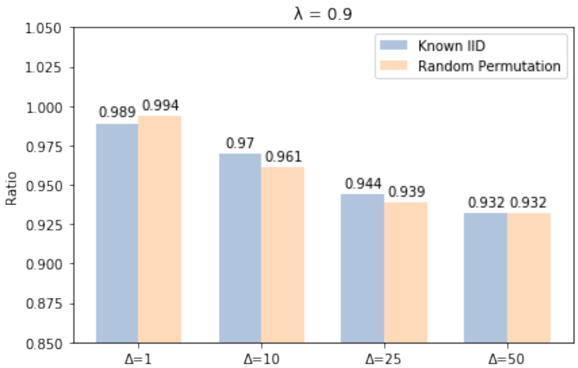

Figure 3.6: Experimental ratios for different values of Δ ($\lambda = 0.9$)

Figures 3.7 - 3.10 present the computation times for different values of the parameters λ and Δ. The average computation times ranged between 0.03 and 1.95 seconds (known IID) and between 0.11 and 7.52 seconds (random permutation). The small value of Δ means that the algorithm takes much computation time because the number of stochastic programming formulations to be solved increases. It is important to decide an appropriate value of Δ by considering the experimental ratios and computation times. Hence, considering a trade-off between the experimental ratios and computation times presented in Figures 3.3 - 3.10, we decided to set λ to 0.3 and Δ to 10 for the following analysis. The empirical results for different ratios of $\frac{U_i}{L_i}$, $|A|$, and $|T|$ are presented in the next subsections.

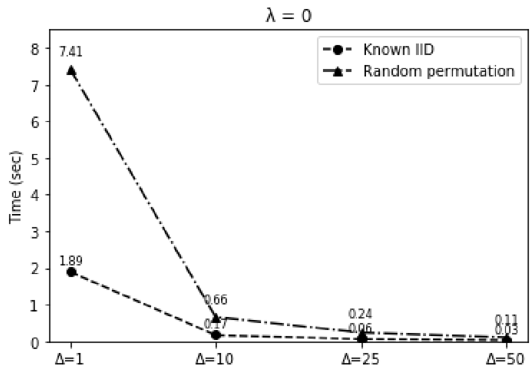

Figure 3.7: Computation times for different values of Δ ($\lambda = 0$)

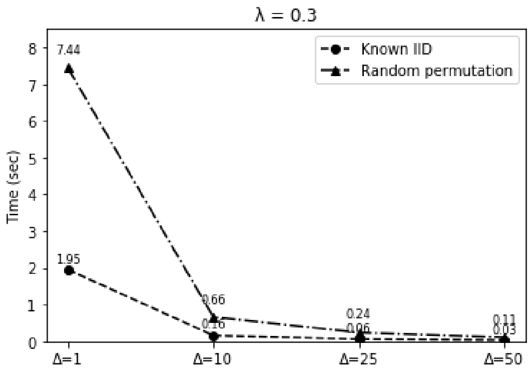

Figure 3.8: Computation times for different values of Δ ($\lambda = 0.3$)

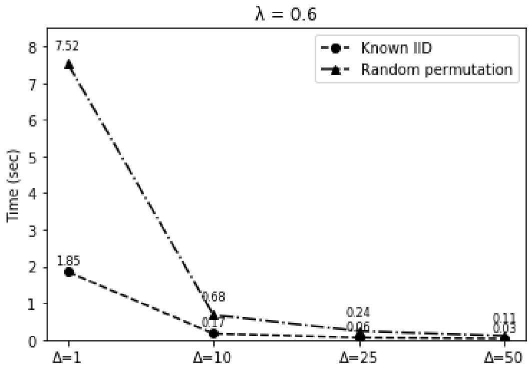

Figure 3.9: Computation times for different values of Δ ($\lambda = 0.6$)

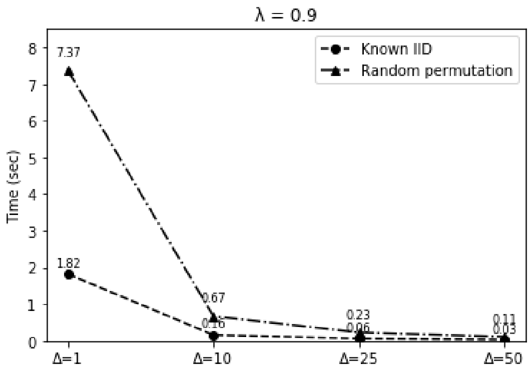

Figure 3.10: Computation times for different values of Δ ($\lambda = 0.9$)

3.4.1 Results for known IID model

The empirical results for different ratios of $\frac{U_i}{L_i}$ (between 10 and 1000) are presented in Figure 3.11. Figure 3.11 shows the average values for 100 data sets. To compare the performances of the primal-dual algorithm with stochastic information (Algorithm 3.1), we present the results for the primal-dual algorithm using a greedy update rule as well. The average differences in experimental ratios between the two algorithms ranged between 10.5% and 14.6%. The difference tended to increase as $\frac{U_i}{L_i}$ increased, but the tendency was not high. The ratio of $\frac{U_i}{L_i}$ considerably affects the worst-case bound obtained by Algorithm 2.3, while it did not highly affect the experimental ratios obtained by Algorithm 3.1 in the known IID model. Regardless of the ratios of $\frac{U_i}{L_i}$ (between 10 and 1000), the experimental ratios of Algorithm 3.1 showed more than 98%.

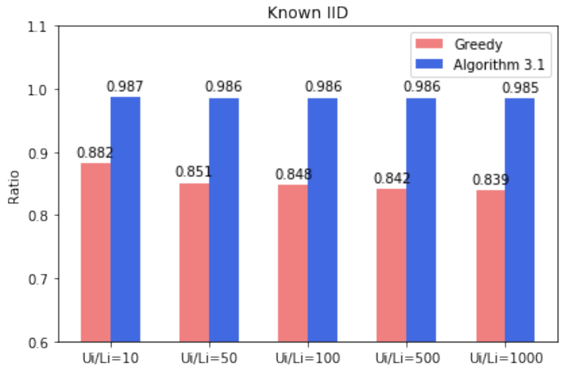

Figure 3.11: Results for different ratios of U_i/L_i (Known IID)

Figure 3.12 presents the experimental ratios for different values of $|A|$ and $|T|$ (6 cases). Figure 3.13 shows the computation times for different values of $|A|$ and $|T|$. The values are average values for 100 data sets under $\frac{U_i}{L_i} = 100$. The experimental ratios tended to slightly increase as $|A|$ and $|T|$ increased in both of the two approaches. This implies that as $|A|$ and $|T|$ increase, advertisements not assigned yet are more likely to have the opportunity to be displayed on the remaining slots. The results from Algorithm 3.1 showed nearly 99% experimental ratio and the difference of $13 \sim 14\%$ compared to the greedy approach in these experiments. The computation times ranged between 0.18 and 9.75 seconds and tended to show the tendency of increasing exponentially.

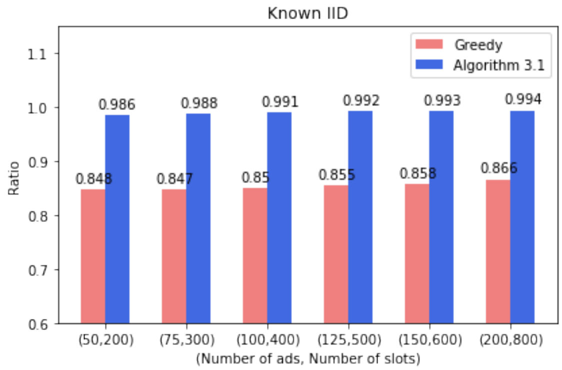

Figure 3.12: Experimental ratios for different values of $|A|$ and $|T|$ (Known IID)

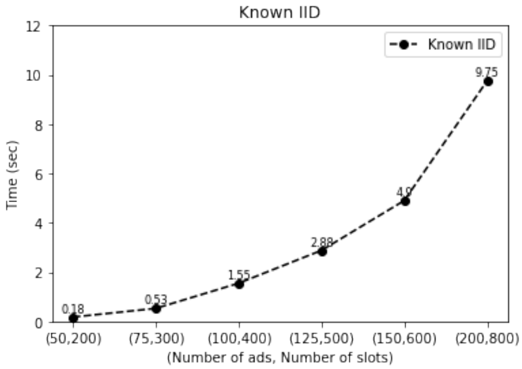

Figure 3.13: Computation times for different values of $|A|$ and $|T|$ (Known IID)

3.4.2 Results for random permutation model

The empirical results for different ratios of $\frac{U_i}{L_i}$ (between 10 and 1000) are presented in Figure 3.14. Figure 3.14 shows the average values for 100 data sets. The average differences in experimental ratios between the two algorithms ranged between 10.5% and 30.1%. The difference tended to increase as $\frac{U_i}{L_i}$ increased. It showed more distinct differences than in the known IID model. Like that for the known IID model, the ratio of $\frac{U_i}{L_i}$ did not highly affect the experimental ratios obtained by Algorithm 3.1 in the random permutation model. Regardless of the ratios of $\frac{U_i}{L_i}$ (between 10 and 1000), the experimental ratios of Algorithm 3.1 showed nearly 99%.

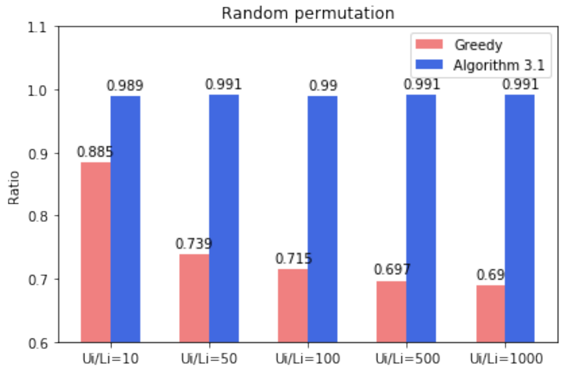

Figure 3.14: Results for different ratios of U_i/L_i (Random permutation)

Figure 3.15 presents the experimental ratios for different values of $|A|$ and $|T|$ (6 cases). Figure 3.16 shows the computation times for different values of $|A|$ and $|T|$. The values are average values for 100 data sets under $\frac{U_i}{L_i} = 100$. The experimental ratios tended to slightly increase as $|A|$ and $|T|$ increased in both of the two approaches. Like that for the known IID model, this implies that as $|A|$ and $|T|$ increase, advertisements that had missed the chance of being displayed on the past slots are more likely to have the opportunity to be displayed on the remaining slots. Algorithm 3.1 showed more than 99% experimental ratios and the difference of $19 \sim 27\%$ compared to the greedy approach in these experiments. The computation times ranged between 0.67 and 102.04 seconds and tended to show the tendency of increasing exponentially. In these experiments, the random permutation model showed more effective results in terms of experimental ratios, but less efficient results in terms of computation times compared to the known IID model.

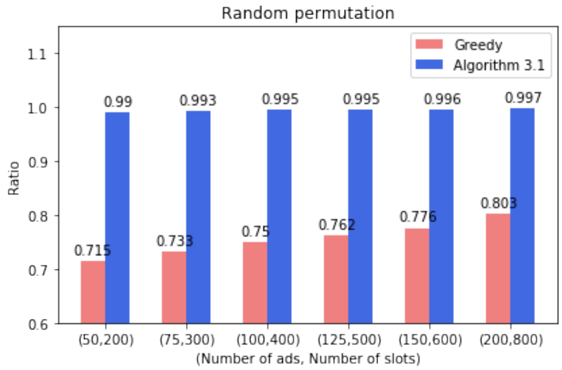

Figure 3.15: Experimental ratios for different values of $|A|$ and $|T|$ (Random permutation)

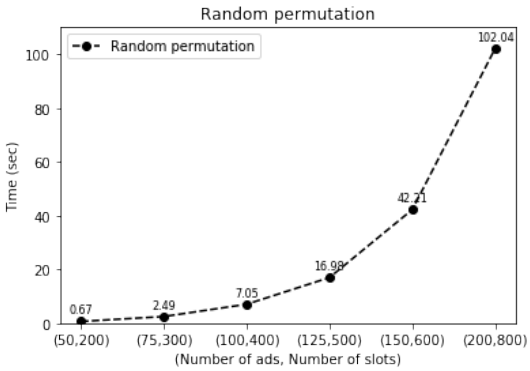

Figure 3.16: Computation times for different values of $|A|$ and $|T|$ (Random permutation)

3.4.3 Managerial insights for Algorithm 3.1

Algorithm 3.1 used in the probabilistic orders has some managerial insights. Algorithm 3.1 can be a robust approach to solving the display ads problem if we know probabilistic information of the coming slots. Overall, Algorithm 3.1 found more than 95% ratios (close to 100% in random permutation) in all cases of the numerical experiments. Even though Algorithm 3.1 did not present guarantee worst-case bounds, we showed that the algorithm is a practical methodology through the simulation results.

There are a lot of future scenarios in each slot. We arbitrarily select sample scenarios ($|\Omega_j|$) and solve the stochastic programming problem with the sample scenarios, because it would take much time to consider all future scenarios. Through the simulation results, we showed more than 97% experimental ratios even in a small number of scenarios ($|\Omega_j| = 4$ or 5). In fact, there is a possibility that a small number of scenarios lead to bias. However, the simulation results found that the bias might not have a significant impact on the experimental ratios.

It is crucial to decide appropriate values of Δ and λ when we use Algorithm 3.1 to solve the problem. The parameter Δ is related to the trade-off between effectiveness and efficiency. The parameter λ is used to adjust the effect of the stochastic technique. We found that using only the adjust dual variable β_i^ω ($\lambda = 0$) does not always lead to better results through the experiments. Thus, it is recommended that publishers decide appropriate values of Δ and λ through their own simulation results.

From a practical point of view, the simulation results from Algorithm 3.1 can apply to a wide range of fields. For example, the publishers would need Algorithm

3.1 to solve the online advertising assignment problem in which some advertisers are sensitive to the number of times their ads are displayed. It would also be helpful to solve other types of problems (e.g., scheduling or resource allocation) that can correspond to the online bipartite matching problem [15, 90]. In these problems, the number of resources (e.g., humans, machines, hotel rooms, and so on) is limited, which has to present a strict capacity constraint.

3.5 Summary

In this chapter, we considered the display ads problem, which is a generalization of the edge-weighted and capacitated online bipartite matching problem as covered in Chapter 2. Unlike Chapter 2, we derived stochastic formulations for the problem by considering two probabilistic order models (known IID and random permutation), which are suitable for the realistic situation. For the probabilistic orders, the stochastic online algorithm with scenario-based stochastic programming and Benders decomposition was proposed to solve the problem.

The stochastic online algorithms provided better performances than the primal-dual algorithm did through the numerical experiments of the two probabilistic orders. The appropriate values of λ and $|\Delta|$ must be chosen according to the trade-off between the experimental ratio and computation time. The solution methodologies provided competitive and realistic solutions in the real-time environment of the assignment problem. Therefore, these algorithms are expected to be useful for publishers who place online advertisements on their websites.

Chapter 4

Online Banner Advertisement Scheduling for Advertising Effectiveness

4.1 Problem Description and Literature Review

Advertising is used as one of the critical methods for companies to promote products or services. There are various platforms to present advertisements, such as broadcast, print, or the Internet. With a drastic increase in online communities, many companies have been paying attention to web advertising. Besides, web advertising has more traceability, cost effectiveness, reachability, and interactivity than other platforms, so the popularity of web advertising can be expected to continue [95]. According to an Internet Advertising Bureau (IAB) report, Internet advertising revenue in the United States was $107.5 billion in 2018, an increase of 22% over that of 2017 [68].

With the advent of web advertising, studies related to web advertising have also been initiated. [122] provided a framework for the competitive selection of advertisements on websites. [106] described terminologies for web advertising measurements and proposed exposure and interactivity metrics. [98] explained the growth of web advertising and proposed a web advertising model for effectiveness.

Banner advertisement is the most common form of web advertising. A banner is a rectangular advertisement positioned on either side, the top, or the bottom of a web page, such as an Internet article or an online shopping mall. An image in the banner advertisement is clickable and is linked to a target web page [98]. By being displayed on a web page, the banner attracts the attention of online users who are interested in the associated products or services.

Online advertisement publishers must construct layout partitioning and allocate the appropriate banners at the respective time [58, 97]. A well-constructed advertisement scheduling increases advertising effectiveness. As the advertising effectiveness increases, advertisers' requests to display their advertisements on the website increase. As the requests increase, online advertisement publishers can ultimately generate more revenue [116, 22]. Hence, online banner advertisement scheduling is important to online advertisement publishers. Online banner advertisement scheduling for online advertisement publishers is addressed in this study.

Many researchers have focused on online banner advertisement scheduling from an optimization perspective and provided mathematical models. [1] presented a MAXSPACE problem of banner advertisement scheduling; it was designed to find a banner advertisement scheduling solution in which space utilization of various-sized advertisements is maximized for specific time slots. Figure 4.1 shows an example solution to the problem (left) and a webpage for the second slot (right) in the planning period. In this example, there are six advertisements (A to F) and ten slots in the planning period. The advertisements each have different lengths, and they can be assigned to each slot. Each slot can be represented as a period of five, ten, thirty minutes, etc. A banner is positioned on the left side of the webpage in this figure.

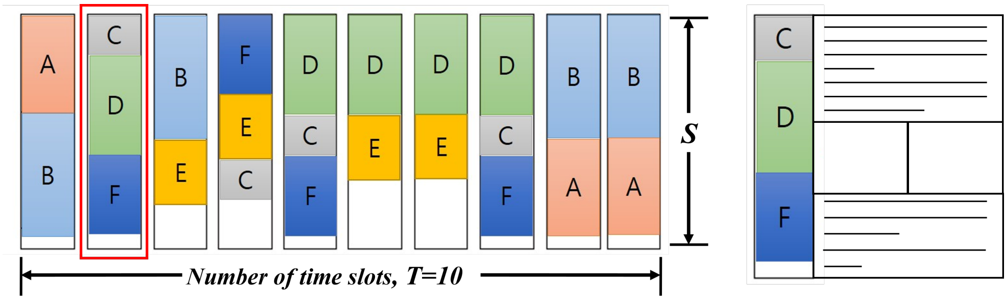

Figure 4.1: Example of MAXSPACE problem (left) and webpage for the second slot (right)

Recent research on online banner advertisement scheduling has focused on maximizing advertising effectiveness, not space utilization by introducing new objective functions [7, 36, 54, 74]. They pointed out that maximizing the space utilization of time slots does not directly translate to maximizing online advertisement publishers' revenue. To measure advertising effectiveness, they primarily took into account the timing when an advertisement is displayed and the number of advertisement exposures. However, they have not extensively investigated the effects of other advertisements displayed in the same time slot. [120, 62, 5] explicitly demonstrated that the effectiveness of any given advertisement might be diminished by the presence of competing advertisements sorted and served together with it in a time slot. Such interference could play an important role, in particular, when the companies display advertisements of their own products or services on the website. Therefore, this study develops an online banner advertisement scheduling model that includes critical factors, such as competitive advertising interference, influencing advertising effectiveness.

In this study, there are four distinct features for the online banner advertisement scheduling considering advertising effectiveness. First, to reflect the advertising effectiveness, this model considers factors that influence the tendency for online users to click on the advertisement. The tendency is defined as the *click-through-rate* (CTR) [55, 93]. This study proposes four factors that influence the CTR: *exposure*, *involvement*, *size*, and *competition*. In addition to these factors, the contents and features of an advertisement (such as message, colour, and animation) also influence the CTR. However, because online advertisement publishers cannot control content features, we do not consider them in this study. That is, this study assumes that the contents of each advertisement are fixed before the banner advertisement scheduling is implemented.

Second, this study is the first one in which competitive advertising interference is introduced to online banner advertisement scheduling. We call the interference 'competition.' Competition is defined as the extent of interference when consumers are exposed to advertisements for competing products, and negatively affects advertising effectiveness [62, 120, 28]. When more advertisements are displayed together in a time slot, each advertisement is adversely affected by the high competition within the slot. So, it may not be optimal to display as many advertisements as possible in a time slot.

Third, this study presents an expected CTR function as an objective function of the model. The function is based on the consideration of the four factors mentioned above. This study uses the demand function developed by [30] to devise the expected CTR function. Each component of the demand function can be interpreted as one of the factors of the expected CTR function. Fourth, variable display frequency bounds proposed by [37] are used in our model. The constraint not only reflects a realistic situation but also offers more flexibility for online advertisement publishers to display advertisements in the model compared to fixed display frequency.

With a given set of advertisements, online advertisement publishers have to design a well-constructed advertisement scheduling model to maximize advertising effectiveness. It leads to an increase in advertisers' requests to display their advertisements on the website and ultimately generates more revenue for online advertisement publishers. The model that considers the four features, as mentioned above, can be represented as an integer programming model with the non-linear objective function. This study not only develops an online banner advertisement scheduling model to maximize advertising effectiveness but also provides a solution methodology to obtain good quality solutions efficiently: a heuristic approach to finding valid lower and upper bounds of the model. To compare the result from large data sets, we also present a hybrid tabu search.

There exists various literature on online banner advertising scheduling and mathematical models. [2] proposed a banner assignment problem by using the formulation of the minimum cost flow problem. [1] presented a MAXSPACE problem of banner advertisement scheduling. They also showed that the problem is NP-hard. A heuristic algorithm called SUBSET-LSLF was developed for obtaining good advertisement scheduling solutions for web pages. [34, 88] presented better heuristic algorithms to solve the MAXSPACE problem than those offered by to [1]. [52] presented a $(3 + \epsilon)$-approximation algorithm for a profit maximization problem (equal to the MAXSPACE problem).

Furthermore, some studies extended the original MAXSPACE problem by considering realistic situations. [11] presented the MAXSPACE problem regarding multiple display frequencies in which the customer is allowed to specify a set of acceptable frequencies. [37] presented the MAXSPACE problem that incorporated variable display frequencies. Online advertisement companies apply acceptable advertisement frequency ranges to offer more flexibility to the publishers and advertisers. [108] proposed an advertisement scheduling problem with different advertisement frequencies between prime and regular times. [38] stated the importance of online advertisement targeting, and [35] extended the problem of [37] by considering advertisement targeting. Some studies proposed placement models and formulations in which two-dimensional display time slots are considered [18, 60, 59, 82, 85, 95]. [33] considered release dates and deadlines in the MAXSPACE problem and developed a polynomial-time approximation scheme for the problem.

Meanwhile, some studies provided new pricing models that reflect applicability in industry. The studies emphasized maximizing advertising effectiveness rather than space utilization. [87] solved the online banner advertisement scheduling problem with a hybrid pricing model. A hybrid pricing model is the one where pricing is based on a combination of the number of advertisement exposures (cost-per-thousand impressions model) and the number of clicks on the advertisement (click-through model). [107, 44, 32] discussed the importance of the position and timing when managers assign advertisements to the slots in advertisement channels, such as online web, mobile device, book, and TV. [36] extended [37]'s model and presented a nonlinear pricing model which reflects the quantity discount pricing strategy.

Also, some studies addressed new pricing models combined with the characteristic of online users. [7] discussed a contract problem for online advertisements with pay-per-view pricing scheme. [42]'s model is to accomplish that each advertisement is displayed as proportionally as possible to all targeted viewers types. [74, 54] considered online users' preference and fatigue regarding online advertising scheduling. [91] presented an online display advertising planning problem to maximize the spreading of impressions across targeted audience segments, while limiting demand shortfalls.

Compared to the previous literature, this study presents a new approach to measuring advertising effectiveness. The approach considers four factors that influence the tendency for online users to click on the advertisement and presents an expected CTR function to reflect these factors. This study first presents the expected CTR function in this problem. Among the factors, the degree to competition has not yet been discussed extensively in the literature on maximizing advertising effectiveness. Because of the degree to competition, the problem may violate the monotonicity assumption. It means that the effectiveness of a set of ads (set A) may be smaller than that of the subset of set A. It may be difficult to solve the problem using dynamic programming or knapsack problem with uncertain/independent values [102, 103, 21, 104]. Hence, we present other solution approaches to solving the problem.

The remainder of the study is organized as follows. Section 4.2 presents descriptions of the assumptions and the formulation of the online banner advertisement scheduling for advertising effectiveness. Section 4.3 presents the solution methodologies used to obtain valid lower and upper bounds of the problem efficiently. In Section 4.4, experimental results for small and large problems are provided. The results using the standard data sets provided by the IAB are also presented. Section 4.5 offers summaries of the research and suggestions for future studies.

4.2 Mathematical Model

In this section, an integer programming model with a non-linear objective function for the online banner advertisement scheduling problem is explained. The objective of the model is to select a set of advertisements for each time slot so that the advertising effectiveness is maximized. A time slot is a space for displaying advertisements in a unit time interval. There are three types of constraints in our model. First, the selected advertisements for a time slot should fit in the available space. Second, each advertisement can be displayed at most one in a time slot. Third, if an advertisement is selected, its display frequency in all time slots must be between lower and upper bounds.

4.2.1 Objective function

Unlike prior models of online banner advertisement scheduling, the model presented herein is based on considerations of advertising effectiveness [37, 82, 95]. Accordingly, the advertising effectiveness is reflected in the objective function of the model. This study proposes an expected CTR function as an objective function of the model. CTR is typically defined as the number of clicks divided by the number of times the advertisement has been exposed to time slots; thus, it represents the effectiveness of an advertisement [16, 55]. The actual value of CTR is calculated after advertisements are displayed on a slot. Publishers, who place advertisements on the slots, need a reasonable estimation of the CTR values to maximize the advertising effectiveness when a set of ads and time slots are given. After estimating the CTR values, publishers can accomplish effective assignments based on the values.

Table 4.1: Four factors that influence the click-through-rate (CTR)

Factors	**Description**	**References**
Exposure	The extent to how often the advertisement is exposed over a planning horizon	[112, 121]
Involvement	The number of website users who have high-affinity with the advertisement or the degree of relationship between the website and the advertisement	[16, 121, 113]
Size	The relative or absolute size of the advertisement	[92, 113]
Competition	The extent of interference when consumers are exposed to advertisements for competition products	[120, 62]

The expected CTR function used in this study consists of four main factors: exposure, involvement, size, and competition. The four factors are not only closely related to CTR but also those that online advertisement publishers can control through changing a subset of advertisements displayed when they display advertisements on websites. Because online advertisement publishers cannot control the contents of an advertisement itself, factors such as message, colour, and animation are not considered. In other words, it is assumed that the contents of each advertisement are fixed. Table 4.1 describes the four factors that compose the expected CTR function and presents previous research which found statistically significant relationships between the four factors and CTR.

This study presents an expected CTR function based on the consideration of the four factors listed in Table 4.1. The function was made by referring to the demand function developed by [30]. The demand function of [30] is defined as follows:

$$D_{ik} \text{ (expected demand for product `}i\text{' displayed on shelf `}k\text{')} = \alpha_i (s_{ik}x_{ik})^{\beta_{ik}} \prod_{\forall j \neq i} \sum_{m=1}^{K} (s_{jm}x_{jk})^{\delta_{ij}} \tag{4.1}$$

The demand function was used to solve the shelf-space allocation problem. (K is the number of shelves.) The demand function was postulated and proved through broad empirical findings [30, 31, 89]. An empirical estimation with cross-sectional data, a meta-analysis, and the method of least square have been used to estimate the parameters of the demand function [39, 43, 45]. The shelf-space allocation problem is used to optimize the retailer's allocation of shelf space for a set of alternative products.

The shelf-space allocation and banner advertisement scheduling are structurally similar. First, the publisher chooses a set of items (products or advertisements) and displays them in limited spaces. Second, the objective is to maximize the publisher's revenue. In addition, the total demand is dependent on the allocation of selected products in the shelf space allocation problem. Likewise, in the online banner scheduling problem, the total advertising effectiveness is dependent on a set of displayed advertisements. Because of the similarities between the two problems, for this study, the expected CTR function was developed by using the components of the demand function. Each component of the demand function can be interpreted as one of the factors of the expected CTR function. In particular, exposure (positive) and competition (negative) effects are clearly reflected in the demand function.

The expected CTR function is presented as follows:

$$\begin{aligned}&\pi_{ik} \text{ (expected CTR for ad `}i\text{' in time slot `}k\text{')} = \\ &\qquad \alpha_i (s_i x_{ik})^{\beta_{ik}} \prod_{\forall j \neq i, x_{jk}=1} (s_j x_{jk})^{\delta_{ij}}\end{aligned} \tag{4.2}$$

The expected CTR function includes the four factors that influence CTR. Table 4.2 shows the comparisons between the demand function developed by [30] and the expected CTR function. The components of [30]'s demand function can be translated into the factors this study proposes. The components of Table 4.2, except x_{ik}, are all parameters. The parameters are assumed to be empirically estimated by historical data for advertising effectiveness. This study uses the expected CTR function as the objective function.

Table 4.2: Comparisons between the demand function developed by [30] and the expected CTR function

	Demand function [30]	**CTR function (this study)**
α_i	Space scale parameter for product i	Quality of contents of ad i
$s_{ik}(s_i)$	Quantity of product i displayed on shelf k	*Size* of ad i
x_{ik}	Product i displayed on shelf k or not	Ad i displayed in slot k or not (*exposure*)
β_{ik}	Space elasticity of product i displayed on shelf k	Degree to *involvement* of users for ad i in slot k
δ_{ij}	Cross space elasticity between products i and j	Degree to *competition* between ads i and j

In shelf-space allocation problem, α_i is the space scale parameter for product i. The value of α_i is translated into the quality of the contents of ad i in the CTR function. The space elasticity is defined as the ratio of changes in sales to changes in space. The value of β_{ik} depends on characteristics of product i and shelf k. Similarly, in the CTR function, β_{ik} represents the degree to involvement of users for ad i in slot k. In other words, β_{ik} emphasizes the importance of timing when ad i is displayed, while α_i means the effect of ad i itself regardless of the timing. The cross-space elasticity, δ_{ij}, is defined as the ratio of changes in sales of product i to changes in product j. The value of δ_{ij} depends on whether products i and j are complementary or substitute products. Similarly, in the CTR function, δ_{ij} represents the degree to competition between ads i and j.

4.2.2 Notations and formulation

The definitions of the parameters and decision variables used in the integer programming model with a non-linear objective function are presented as follows:

Table 4.3: Parameters and decision variables

N	Number of advertisement types
T	Number of time slots
H	Height of banner in all time slots
s_i	Height of ad i
L_i	Lower bound on display frequency of ad i if displayed
α_i	Quality of contents of ad i
β_{it}	Degree to involvement and tendency of users of ad i for time slot t
δ_{ij}	Degree to competition between ads i and j
x_{ij}	Binary decision variable, whose value is 1 if ad i is assigned to time slot t; 0 otherwise
y_i	Binary decision variable, whose value is 1 if ad i is assigned; 0 otherwise

The integer programming formulation with a non-linear objective function for online banner advertisement scheduling is

$$\max \sum_{i=1}^{N} \sum_{t=1}^{T} \pi_{it} x_{it} \tag{4.3}$$

$$\pi_{it} = \alpha_i (s_i)^{\beta_{it}} \prod_{\forall j \neq i,} (1 - x_{jt} + s_j x_{jt})^{\delta_{ij}} \tag{4.4}$$

$$\sum_{i=1}^{N} s_i x_{it} \leq H, \ \forall t \tag{4.5}$$

$$L_i y_i \leq \sum_{t=1}^{T} x_{it} \leq T y_i, \ \forall i \tag{4.6}$$

$$x_{it} \in \{0, 1\}, \ \forall i, t \tag{4.7}$$

$$y_i \in \{0, 1\}, \ \forall i \tag{4.8}$$

The objective of the formulation is to find the optimal banner advertisement scheduling that maximizes advertising effectiveness. The objective function means the total expected CTR value that indicates advertising effectiveness. Equation (4.4) defines π_{it} as an expected CTR value for ad i in time slot t. Constraint (4.5) guarantees that the sum of the heights for advertisements displayed in each time slot cannot exceed the banner height. Constraint (4.6) enforces frequency bounds for each advertisement. For any advertisements displayed, the number of times that the advertisement is shown should be between the lower bound, L_i, and the number of time slots on the time horizon. Constraints (4.7) and (4.8) define x_{it} and y_i as binary variables, respectively.

4.3 Solution Methodologies

The optimization model presented in Section 4.2 featured a non-convex objective function, which reflects the effectiveness of advertising, with linear constraints. Therefore, in a large data set, whose optimal solution can be intractable to compute, the problem might not be solved directly using optimization solvers within a reasonable time. This study explains an alternate algorithm proposed to solve the problem efficiently and effectively. The alternate algorithm finds valid lower and upper bounds through the optimization model. Because optimization solvers might not provide good bounds or solutions within a reasonable time, a hybrid tabu search, which is a meta-heuristic approach, is also presented as a means to compare the results from large data sets.

4.3.1 Heuristic approach to finding valid lower and upper bounds

The heuristic approach focuses on finding valid lower and upper bounds by using the properties of the optimization model (an integer programming model with a non-linear objective function) as presented in Section 4.2. The way to find valid lower and upper bounds is presented as follows:

[**Upper bound**]

The objective function of the optimization model can be divided into two parts. The two parts are called an *involvement part* $\big(\alpha_i \cdot (s_i x_{ik})\big)$ and a *competition part* $\big(\prod_{\forall j \neq i,}(1-x_{jt}+s_j x_{jt})^{\delta_{ij}}\big)$. As the number of advertisements displayed in a time slot increases, the advertising effectiveness, as caused by the involvement part, positively increases; the advertising effectiveness caused by the competition part adversely increases. In other words, for the latter, the value of $\prod_{\forall j \neq i,}(1-x_{jt}+s_j x_{jt})^{\delta_{ij}}$ is always in the range of $(0,1]$, and the value decreases as the competition effect increases. In addition to this fact, the following proposition could be made:

For specific time slot $t \in \{1,2,\dots,T\}$ and set of advertisements $A = \{1,2,\dots,N\}$, let A_1 be subsets of set A such that $n(A_1) = k$ assigned to the time slot t, and A_2 be subsets of set A such that $n(A_2) = k+1$ assigned to the time slot t $(k = 1,2,\dots,N-1)$. Let V_{MAX} be the maximum value of $\sum_{i=1}^{N} \pi_{it} x_{it}$ among the entire subset A_1, and V_{MIN} be the minimum value of $\sum_{i=1}^{N} \pi_{it} x_{it}$ among the entire subsets A_2.

Proposition 4.1. *If $V_{MAX} < V_{MIN}$, then any solutions for which k advertisements are displayed in the time slot t is not an optimal solution.*

Proof. To get a contradiction, assume that let $X^* = (x^*_{11}, x^*_{12}, \dots, x^*_{NT})$ be an optimal solution such that k advertisements are displayed in the time slot t. Let $\hat{X}$ be a feasible solution identical to X^* except for the time slot t in which $k+1$ advertisements are displayed. Let $\Pi(X^*)$ and $\Pi(\hat{X})$, respectively, be the objective values of the problem using X^* and $\hat{X}$. Suppose that $\Pi(X^*) = V_{MAX} + C$. We know $\Pi(\hat{X}) \geq V_{MIN} + C$. Because $V_{MAX} < V_{MIN}$, the outcome is $\Pi(\hat{X}) \geq \Pi(X^*)$. This contradicts the fact that X^* is the optimal solution. $\square$

In summary, as the number of advertisements displayed in a banner increases, the competition effect negatively increases. Therefore, banners with as many advertisements as possible may not always lead to desirable outcomes. If it is guaranteed that one of the solutions with more than k advertisements displayed in a particular time slot is optimal, at least the competition effect in the time slot of the optimal solution is greater than the minimum competition effect when any $k+1$ advertisements are displayed. In the following lemma, an upper bound of the optimization model is obtained. In Lemma 4.2, the value of P is used to determine the range of w. If the N advertisements are sorted in descending order of size, then we define $\overline{S}_i$ as the size of the i^{th} advertisement. The value of P is now defined as the maximum integer value satisfying $\sum_{i=1}^{P} \overline{S}_i \leq H$.

For specific time slot $t \in \{1, 2, \ldots, T\}$ and ad $i \in \{1, 2, \ldots, N\}$, two values (L_{it}^w, R_{it}^w) are defined as follows ($w \in \{2, 3, \ldots, P\}$):

$$L_{it}^w = \begin{cases} \alpha_i(s_i)^{\beta_{it}}, & w = 2 \\ \alpha_i(s_i)^{\beta_{it}} \times Q_1^{max} + (Q_2^{max} \times Q_3^{max}) \times (w-2), & w \neq 2 \end{cases} \tag{4.9}$$

$$R_{it}^w = \alpha_i(s_i)^{\beta_{it}} \times Q_1^{min} + (Q_2^{min} \times Q_3^{min}) \times (w-1) \tag{4.10}$$

where Q_1^{max} is defined as the maximum value of the competition part for ad i when any $w-1$ advertisements, including ad i, are displayed in a slot. Q_1^{min} is defined as the minimum value of the competition part for ad i when any w advertisements, including ad i, are displayed in a slot.

$Q_1^{max} = max\big((s_{j_1})^{\delta_{ij_1}}(s_{j_2})^{\delta_{ij_2}}\cdots(s_{j_{w-2}})^{\delta_{ij_{w-2}}}\big)$ and

$Q_1^{min} = min\big((s_{j_1})^{\delta_{ij_1}}(s_{j_2})^{\delta_{ij_2}}\cdots(s_{j_{w-1}})^{\delta_{ij_{w-1}}}\big)$, $\forall j_1, j_2, \ldots j_{w-1} \in \{1, 2, \ldots, N\}, j_1 \neq j_2 \neq \cdots \neq j_{w-1} \neq i$. Q_2^{max} is defined as the maximum value of the involvement part for an advertisement. Q_2^{min} is defined as the minimum value of the involvement part for an advertisement. That is, $Q_2^{max} = max\ \alpha_{i'}(s_{i'})^{\beta_{i't}}$ and $Q_2^{min} = min\ \alpha_{i'}(s_{i'})^{\beta_{i't}}$, $\forall i' \in \{1, 2, \ldots, N\}, \forall t \in \{1, 2, \ldots, T\}$. Q_3^{max} is defined as the maximum value of the competition part for an advertisement when any $w-1$ advertisements are displayed in a slot. Q_3^{min} is defined as the minimum value of the competition part for an advertisement when any w advertisements are displayed in a slot.

$Q_3^{max} = max\big((s_{j_1})^{\delta_{i'j_1}}(s_{j_2})^{\delta_{i'j_2}}\cdots(s_{j_{w-2}})^{\delta_{i'j_{w-2}}}\big)$ and

$Q_3^{min} = min\big((s_{j_1})^{\delta_{i'j_1}}(s_{j_2})^{\delta_{i'j_2}}\cdots(s_{j_{w-1}})^{\delta_{i'j_{w-1}}}\big)$, $\forall i' \in \{1, 2, \ldots, N\}$,

$\forall j_1, j_2, \ldots j_{w-1} \in \{1, 2, \ldots, N\}$, $j_1 \neq j_2 \neq \cdots \neq j_{w-1} \neq i'$.

For each ad i and time slot t, let w'_{it} be the smallest integer of w such that $L_{it}^w > R_{it}^w$. Using the value of w'_{it}, the value of D_{it}, for each i and t, can be calculated as follows:

$$D_{it} = \begin{cases} 1, & w'_{it} = 2 \\ max\big((s_{j_1})^{\delta_{ij_1}}(s_{j_2})^{\delta_{ij_2}}\cdots(s_{j_{w'_{it}-2}})^{\delta_{ij_{w'_{it}-2}}}\big), & w'_{it} \neq 2 \end{cases} \tag{4.11}$$

If w'_{it} and D_{it} are obtained, then an upper bound (UB) of the optimization model can be calculated. UB is an optimal objective value of a new formulation. The objective function of the new formulation is $max \sum_{i=1}^{N} \sum_{t=1}^{T} \alpha_i(s_i)^{\beta_{it}} x_{it} \cdot D_{it}$ instead of $\sum_{i=1}^{N} \sum_{t=1}^{T} \pi_{it} x_{it}$.

Lemma 4.2. *The objective value of the new formulation* ($max \sum_{i=1}^{N} \sum_{t=1}^{T} \alpha_i(s_i)^{\beta_{it}} x_{it} \cdot D_{it}$ *subject to* (4.5) - (4.8)) *can be an upper bound of the online banner advertising scheduling problem. Also, the new formulation can be represented as binary integer programming (BIP).*

Proof. The values of L_{it}^{w} and R_{it}^{w} can be interpreted as described in this paragraph. For ad i and time slot t, the value of $\sum_{i=1}^{N} \sum_{t=1}^{T} \pi_{it} x_{it}$ is less than L_{it}^{w} when any $w-1$ advertisements including ad i are displayed in the time slot t, and the value of $\sum_{i=1}^{N} \sum_{t=1}^{T} \pi_{it} x_{it}$ is greater than R_{it}^{w} when any w advertisements including ad i are displayed in the time slot t. So, if $L_{it}^{w} \le R_{it}^{w}$, then the solutions with fewer than w advertisements (including ad i) displayed in the time slot t are guaranteed to be not optimal. In other words, if w_{it}' is obtained for ad i and time slot t, the solutions with fewer than $w_{it}' - 1$ advertisements (including ad i) displayed in the time slot t are not optimal. Furthermore, D_{it} represents the value of the competition part that reflects the minimum competition effect when $w_{it}' - 1$ (including ad i) advertisements are displayed in time slot t. Thus, for each ad i and time slot t, the competition effect of the optimal solution is not less than the competition effect indicated by D_{it}. Accordingly, for the new formulation, in which the competition part $\left(\prod_{\forall j \neq i,} (1 - x_{jt} + s_j x_{jt})^{\delta_{ij}}\right)$ is replaced by D_{it}, the objective value is guaranteed to be an upper bound of the original model. The new formulation can be a BIP problem as follows: $UB = max \sum_{i=1}^{N} \sum_{t=1}^{T} \alpha_i(s_i)^{\beta_{it}} x_{it} \cdot D_{it}$ subject to (4.5) - (4.8). $\square$

The procedure of the way to obtain a lower bound is as follows:

[**Lower bound**]

The optimization model can be a BIP if no competition effects are present ($\forall \delta_{ij} = 0$). In Stage 1, the heuristic solves the optimization model without competition effects (BIP). In the next iterations (Stage 2), we assume that the competition part for each advertisement is calculated by using X^* obtained from the previous iteration. We then update the objective function and solve the corresponding BIP. The heuristic to obtain a competitive lower bound is a modification of the heuristic of [114]. The details of the heuristic are as follows:

Stage 1. Solving the optimization problem without competition effects.

Step 1. Let X^* be the solution of the optimization problem without competition effects.

Step 2. $LB \leftarrow$ the objective function value applying X^* to the original formulation.

Stage 2. Solving the optimization problem with competition effects of the previous iteration.

Step 1. For $\forall i, t$, compute π_{it} using X^* and construct the corresponding BIP using X^*.

Step 2. Let X^{**} be the solution of the corresponding BIP using π_{it}

Step 3. If $X^* = X^{**}$, then stop the process; otherwise, go to Step 4.

Step 4. $LB^* \leftarrow$ the objective function value applying X^{**} to the original formulation.

Step 5. If $LB^* > LB$, then $LB \leftarrow LB^*$.

Step 6. $X^* \leftarrow X^{**}$ and repeat Stage 2.

4.3.2 Hybrid tabu search

This study uses a hybrid tabu search as a supplementary approach to compare the results from large data sets. A tabu search is a meta-heuristic method that can be used to explore a space of possible solutions beyond the local optimality within a reasonable time [56]. The method is easy to represent a feasible solution as a sequence form in this problem and also to find an improved solution by checking the immediate neighbor sequences of the current solution. Thus, the hybrid tabu search is used as a supplementary approach. The hybrid tabu search generates an initial sequence based on a greedy algorithm, and then the tabu search is repeated with the updated sequence. The sequence represents the assignment of advertisements to time slots. When constraints are satisfied, the sequence can be converted to a feasible solution.

Appendix A presents the details of the greedy algorithm that generates an initial sequence and its information for the hybrid tabu search. After the greedy algorithm, we conduct the hybrid tabu search using this sequence to obtain good feasible solutions. The details of the hybrid tabu search are presented in Appendix B (Algorithm B.1 and Algorithm B.2). $num1$ is an arbitrary constant that represents a criterion for assigning advertisements, which is introduced to avoid being trapped in local optima. A small example is presented in Appendix C.

4.4 Computational Experiments

In this section, the performances of the model are analyzed in small, large, and standard data sets. The standard data used in this study is extracted from the specifications provided by the IAB. The optimal solution of the non-convex and non-linear formulation was obtained by using a brute-force search with LINGO version 17.0. The heuristic approach used to find valid lower and upper bounds was run with Xpress Mosel version 3.10.0, and the hybrid tabu search was run with JAVA language in Windows 7 on a PC with an Intel(R) Core(TM) i5-4690 CPU 3.5GHz with 16.00 GB of RAM.

Table 4.4: Parameter sets

Parameter	**Value**
(N, T) (small)	(4,2), (5,2), (5,3), (6,3), (6,4), (7,3)
(N, T) (large)	(10,3), (20,5), (30,10), (50,15), (75,20), (100,25)
H	400
L_i	1 or 2 (small); 1,2,3 (large)
s_i	U[80, 100]
α_i	U[5.0, 10.0]
β_{it}	U[0.10, 0.20]
δ_{ij}	U[-0.020, -0.010]

For small data sets, the performances of the solution methodologies were evaluated by comparing the optimal solutions obtained within a reasonable time. The time limit was set to 3,600 seconds. However, for large data sets, because the formulation could not provide competitive solutions as well as an optimal solution within the time limit, the results obtained by the heuristic approach and hybrid tabu search only were analyzed. The parameters of the hybrid tabu search are as follows: $num1$ was set to 0.05, the number of iterations was $10 \times N$, and the number of tabu search iterations was 5. Table 4.4 shows the parameter sets used in the formulation. U$[a, b]$ refers to a uniform distribution between a and b.

The parameters N and T in small data sets were designed to solve the problem using the brute-force search easily and those in big data sets were chosen with the rate of approximately 3 or 4 to 1 (N:T). The slot size and advertisement size were chosen to display an average of 4 or 5 advertisements in a time slot. The ranges for the parameters related to demand function of [30], such as α_i, β_{it}, and δ_{ij} were based on values from shelf-space allocation problems. The results for small and large data sets are presented in Sections 4.4.1 and 4.4.2. In Section 4.4.3, the computational results, using the standard data sets developed by [66], are presented.

4.4.1 Results for problems with small data sets

Twenty different samples were tested for each problem size. The brute-force search could find the optimal solutions in all instances of small data sets within the time limit. However, the problems that exceeded the (7, 3) instance set did not provide near-optimal solutions as well as an optimal solution within the time limit. The sample data were also used in the other solution methodologies to compare the results in terms of the effectiveness and efficiency. Tables 4.5 and 4.6 show the results for the problems with small data sets.

Table 4.5: Computation times for problems with small data sets

(N, T)	**Computation time (in seconds) (avg, max)**		
	Optimization	Heuristic for bounds	Hybrid tabu search
(4, 2)	(18.8, 23.0)	(0.025, 0.028)	(0.010, 0.014)
(5, 2)	(52.5, 78.0)	(0.027, 0.035)	(0.014, 0.017)
(5, 3)	(109.6, 230)	(0.020, 0.033)	(0.020, 0.026)
(6, 3)	(256.1, 480)	(0.030, 0.056)	(0.030, 0.039)
(6, 4)	(454.9, 766)	(0.030, 0.058)	(0.059, 0.090)
(7, 3)	(1,076, 2,473)	(0.050, 0.060)	(0.040, 0.050)

Table 4.6: Performance gaps for problems with small data sets

(N, T)	% performance gap (avg, max)		
	Heuristic for bounds*	Hybrid tabu search*	$1 - \frac{LB}{UB}$
(4, 2)	(0.00, 0.00)	(0.00, 0.00)	(0.00, 0.00)
(5, 2)	(0.11, 0.46)	(0.02, 0.24)	(1.48, 2.00)
(5, 3)	(0.14, 0.50)	(0.13, 0.50)	(1.47, 2.16)
(6, 3)	(0.28, 0.88)	(0.21, 0.60)	(2.32, 3.39)
(6, 4)	(0.27, 0.72)	(0.20, 0.44)	(2.33, 3.15)
(7, 3)	(0.40, 0.71)	(0.18, 0.54)	(2.83, 3.36)

* $\left(1 - \frac{\text{objective value of solution obtained by each algorithm}}{\text{optimal objective value}}\right) \times 100\%$

The average computation times using the brute-force search ranged between 18.8 and 1,076 seconds. In contrast, the average computation times for the other two solution methodologies were less than 0.05 seconds. The heuristic approach to finding valid lower and upper bounds found the optimal solutions in 44 of 120 instances, and the hybrid tabu search found the optimal solutions in 58 of 120 instances. Although the optimal solutions were not obtained in more than one-half the instances, the worst optimality gaps for 120 instances were 0.88% and 0.60% for the heuristic approach and the hybrid tabu search, respectively. On average, for the small data sets, the hybrid tabu search performed slightly better and found the solution a little faster than the heuristic approach. In the heuristic approach, the average percentage of $(1 - \frac{LB}{UB})$ was calculated to be 1.74%, and the worst percentage was 3.39%. The average gap between the upper bound and the optimal objective value was calculated to be 1.53%, and the worst gap was 3.16%. The two solution methodologies found comparatively good quality solutions within one second in the experiments with small data sets. The results showed that the two methodologies are appropriate approaches in terms of effectiveness and efficiency.

4.4.2 Results for problems with large data sets

In this section, the results for the problems with large data sets are presented. Because optimal solutions were not able to be intractable to compute within the time limit by using the brute-force search, the results were analyzed by using the bounds that the heuristic found and the feasible solutions obtained by the hybrid tabu search. In this problem, the number of tabu search iterations was set to T. Each time Algorithm B.2 was executed, $5 \times N$ times were repeated because Algorithm B.2 has a random setting in the procedure. The number of tabu sequences was set to be equal to the number of tabu search iterations. Twenty different samples were tested for each problem size. Tables 4.7 and 4.8 show the results for the problems with large data sets. The last column of Table 4.8 lists the optimality gap indicated in the LINGO solver at 3,600 seconds.

Table 4.7: Computation times for problems with large data sets

(N, T)	**Computation time (in seconds) (avg, max)**	
	Heuristic for bounds	Hybrid tabu search
(10, 3)	(0.05, 0.08)	(0.07, 0.09)
(20, 5)	(0.14, 0.39)	(0.86, 0.96)
(30, 10)	(0.64, 1.32)	(7.55, 7.71)
(50, 15)	(2.55, 5.55)	(54.0, 54.6)
(75, 20)	(9.93, 18.4)	(251, 255)
(100, 25)	(23.7, 46.4)	(799, 812)

Table 4.8: Performance gaps for problems with large data sets

(N, T)	% performance gap (avg, max)				
	Heuristic for bounds*	Hybrid tabu search*	$1-(LB/UB)$	Piecewise linear.	Optimality gap
(10, 3)	(0.25, 0.96)	(0.04, 0.79)	(4.51, 5.56)	(17.3, 18.5)	(26.7, 31.3)
(20, 5)	(0.18, 0.83)	(0.16, 0.83)	(8.73, 9.19)	(17.1, 17.6)	(48.4, 85.2)
(30, 10)	(0.06, 0.31)	(0.56, 2.35)	(8.97, 9.84)	(17.2, 18.5)	-
(50, 15)	(0.00, 0.00)	(1.42, 2.84)	(8.35, 9.49)	(16.4, 17.2)	-
(75, 20)	(0.00, 0.02)	(2.29, 3.77)	(7.70, 9.58)	(15.6, 17.2)	-
(100, 25)	(0.00, 0.00)	(4.57, 15.0)	(7.61, 10.2)	(16.1, 17.9)	-

* $\left(1 - \frac{\text{objective value of solution obtained by each algorithm}}{\max\{LB,\text{objective value obtained by hybrid tabu search}\}}\right) \times 100\%$

The average computation times ranged between 0.05 and 46.47 seconds in the heuristic approach and between 0.06 and 812.75 seconds in the hybrid tabu search. As N and T increased, the heuristic approach found better feasible solutions than the hybrid tabu search did. Specifically, in the (100, 25) instances, the heuristic approach found the feasible solution for which the objective value was higher than that of the hybrid tabu search by 15.02%. On average, the heuristic approach performed better and found the solution faster than the hybrid tabu search did for the large data sets. The results showed that the heuristic approach performed better as N and T increased. In the heuristic approach, the average percentages of $(1 - \frac{LB}{UB})$ ranged between 4.51% and 8.97% for the large data sets. It was not always shown that the average percentages of $(1-\frac{LB}{UB})$ was increased as N and T increased in the instances.

A linearization technique for bilinear programming problems was also used and analyzed to relax the non-linear and non-convex objective function in this problem

[69]. Appendix D presents the details of the linearization technique for bilinear programming problems. When the linearization technique was applied, it showed the same results as the results without competition effects. The average gap from the technique ranged between 15.4% and 17.3% for the large data sets. As in the heuristic approach, the average gap did not always increase as N and T increased in the instances.

The last column of Table 4.8 shows the gap between the best possible objective value and best upper bound (optimality gap) indicated in the LINGO solver at 3,600 seconds. In the (10, 3) and (20, 5) instances, the average optimality gap was calculated to be 26.7% and 48.4%, respectively. The brute-force search could not obtain any feasible solutions during 3,600 seconds for most of the data between (30, 10) and (100, 25). Although the brute-force search obtained some feasible solutions during 3,600 seconds for a few data between (30, 10) and (100, 25), the optimality gaps were more than 80%. The reduction of the optimality gap was less than 3% on average, even when the time limit was changed from 3,600 seconds to 10,800 seconds. Accordingly, the results of the linearization technique and brute-force search support the need for alternate algorithms for online banner scheduling to maximize advertising effectiveness in terms of solution quality and computation time. In these experiments, the heuristic approach to finding valid upper and lower bounds was better than the hybrid tabu search in terms of effectiveness and efficiency.

4.4.3 Results for problems with standard data

In this section, the numerical results using standard data sets are presented. The standard advertisement specification was developed by [66]. Because of the increase in the use of mobile devices and tablet PCs as well as diversification of the types of advertisements, the size specifications for advertisements have consecutively changed recently. Moreover, flexible-sized advertisement units tend to be provided for each type of advertisement. This study focused on a vertical banner with advertisement sizes and other parameters from [37] to deal with the standard data. The advertisement sizes used are described by the IAB.

For vertical banners, the height of a banner is 900 (width is 120) with advertisement sizes of $120 \times 60, 120 \times 90, 120 \times 150, 120 \times 200, 120 \times 240$, and 120×280. The values of α_i, β_{it}, and δ_{ij} were the same as those featured in Sections 4.4.1 and 4.4.2. The unit for the time slot was 5 minutes. The unit for the time slot was set larger than that found in previous literature; otherwise, the difference of β_{it} would be too small over time. Twenty different samples were tested for each problem set.

Table 4.9 shows the parameter sets described in Section 4.4.3, and Table 4.10 presents the results for the problems with the standard data sets. The fourth column of Table 4.9 (number of advertisement sizes) denotes one value of 3, 4, 5, or 6. ($3 = \{120\times200, 120\times240, 120\times280\},\ 4 = \{120\times150, 120\times200, 120\times240, 120\times280\},\ 5 = \{120 \times 90, 120 \times 150, 120 \times 200, 120 \times 240, 120 \times 280\},\ 6 = \{120 \times 60, 120 \times 90, 120 \times 150, 120 \times 200, 120 \times 240, 120 \times 280\}$). Feasible solutions obtained by the heuristic approach to finding valid lower and upper bounds were greater than or equal to those obtained by the tabu search in almost all the standard data sets. In $60T\_5N$ data set, the feasible solution gaps between the heuristic approach and hybrid tabu search were approximately 1%, but in $120T\_15N$ data set, the feasible solution gaps were approximately 10%.

Table 4.9: Parameter sets for problems with standard data sets

	Parameter set			
	T	N	# of ad sizes	L_i
$60T\_5N\_3$	60 (5h)	5	3	35
$60T\_5N\_4$	60 (5h)	5	4	35
$60T\_5N\_5$	60 (5h)	5	5	35
$60T\_5N\_6$	60 (5h)	5	6	35
$60T\_10N\_3$	60 (5h)	10	3	25
$60T\_10N\_4$	60 (5h)	10	4	25
$60T\_10N\_5$	60 (5h)	10	5	25
$60T\_10N\_6$	60 (5h)	10	6	25
$120T\_7N\_3$	120 (10h)	7	3	40
$120T\_7N\_4$	120 (10h)	7	4	40
$120T\_7N\_5$	120 (10h)	7	5	40
$120T\_7N\_6$	120 (10h)	7	6	40
$120T\_15N\_3$	120 (10h)	15	3	30
$120T\_15N\_4$	120 (10h)	15	4	30
$120T\_15N\_5$	120 (10h)	15	5	30
$120T\_15N\_6$	120 (10h)	15	6	30

Table 4.10: Results for problems with standard data sets

	Result (average)	
	$1-(LB/UB)(\%)$	Computation time (s)
$60T\_5N\_3$	2.07	0.06
$60T\_5N\_4$	2.68	0.06
$60T\_5N\_5$	4.05	0.06
$60T\_5N\_6$	7.24	0.05
$60T\_10N\_3$	5.31	0.82
$60T\_10N\_4$	10.54	0.56
$60T\_10N\_5$	14.95	0.79
$60T\_10N\_6$	18.84	0.42
$120T\_7N\_3$	4.32	0.20
$120T\_7N\_4$	7.70	0.29
$120T\_7N\_5$	11.04	0.24
$120T\_7N\_6$	14.64	0.22
$120T\_15N\_3$	10.65	1.20
$120T\_15N\_4$	15.63	1.11
$120T\_15N\_5$	17.45	2.16
$120T\_15N\_6$	23.61	1.32

These problems had complex non-linear objective functions, so the brute-force search requires quite a lot of time to solve the problem or even to find near-optimal solutions. In contrast, the heuristic approach, using Xpress Mosel version 3.10.0, was much more efficient at finding good quality solutions. In most cases, the average computation times for the data sets were less than one second. The last four data sets required slightly more than one second to obtain lower and upper bounds. Because optimal solutions of the problems could not be obtained within the time limit by using the brute-force search, the average ratios of 1−(LB/optimal objective value) and 1−(optimal objective value/UB) were not calculated.

The average ratios of $1 - \frac{LB}{UB}$ ranged between 2.07% and 23.61% in this experiment. For example, in $60T\_5N\_3$ data set, the objective values of the solutions obtained by the heuristic showed at least 97.93% of the optimal solutions on average. The average ratio of $1 - \frac{LB}{UB}$ could be worse (larger) by increasing N, T, and the number of advertisement sizes. In particular, the ratio was influenced most by the number of advertisement sizes. As the number of advertisement sizes was increased, the difference between the biggest and the smallest advertisement sizes in the set was more likely to increase. If the difference between the biggest and the smallest advertisement sizes is big, there is a high possibility that the value w'_{it}, which was used to obtain upper bounds of the problems, is not big. This finding means that the possibility of getting a tighter upper bound is low. It means that tighter bounds and near-optimal solutions can be obtained by using the heuristic approach when the difference between the biggest and the smallest advertisement sizes is small.

4.4.4 Managerial insights for the results

The findings of the problem, the heuristic approach, and the analysis have managerial insights for online advertisement publishers. First, it is not always good to assign as many advertisements as possible in a slot. As the number of advertisements displayed in a time slot increases, the advertising effectiveness might decrease due to competitive advertising interference. The decrease in the expected CTR caused by the competition might be similar to the value of $1-\frac{LB}{UB}$ in the numerical experiments.

Publishers' revenues can be affected along with decreasing CTR value. The gap ranged between about 2% and 23% in the experiments using the standard data sets. The gap was more likely to be larger as the number of advertisements and time slots in a planning horizon increased. It would be more important to assign advertisements in terms of advertising effectiveness as the number of advertisements and time slots increases.

Second, it is important to consider not only the degree to involvement but also the degree to competition. Publishers often try to prioritize and assign the advertisements that have a high degree to involvement in a specific time slot. However, the results showed that they also need to consider the degree to competition between the advertisements in the slot. Thus, it is recommended that publishers choose and schedule advertisements in such a way as to retain the positive effect caused by the degree to involvement while avoiding the negative effect caused by the degree to competition.

Third, unlike in the MAXSPACE problem, the expected CTR values from two groups of advertisements assigned in a slot can be different even when the total lengths of the two groups are the same. Also, the value can be different depending on what slot the group is assigned to. Therefore, it is recommended that publishers analyze the solutions obtained by the heuristic approach, and then identify groups of advertisements that will maximize the degree to involvement and minimize the degree to competition. This approach will help publishers understand the logic of how to assign advertisements and maximize advertising effectiveness.

4.5 Summary

In this chapter, the study contributed to the literature on online banner advertisement scheduling models by considering critical factors influencing advertising effectiveness. In particular, the degree to competition has not yet been discussed extensively in online banner advertisement scheduling. The expected CTR function was devised to reflect the factors in the model and used as an objective function of the model in terms of measuring advertising effectiveness.

The presented model was an integer programming model with a non-linear and non-convex objective function. In a large data set, whose optimal solution can be intractable to compute, the problem might not be solved directly using optimization solvers within a reasonable time. The heuristic approach presented provided competitive solutions efficiently, even for large data sets of non-convex and non-linear models through the properties of the objective function. Also, the problem was tested with standardized conditions for advertisements and time slots, and the heuristic approach performed efficiently and effectively in the standard data sets. The model and its heuristic approach are expected to be useful for online advertisement publishers when they choose and schedule advertisements over a planning horizon while considering advertising effectiveness.

Chapter 5

Conclusions and Future Research

Managing online advertising assignment is essential for publishers who place online advertisements on their websites. Online advertising assignment is generally decided through contracts. In most cases, advertisers and publishers achieve contracts by using compensation methods (e.g., CPC) based on customer performance. For this reason, publishers need to develop their decision-making processes for assigning online advertisements on their website to increase the number of clicks from the users. In this dissertation, we covered two online advertising assignment problems that publishers solve: the display ads problem (one of the search engine advertising problems) and the online banner advertisement scheduling problem (one of the display advertising problems). Moreover, we considered realistic constraints in the problems. Unlike previous assignment problems, the problems might be pragmatic approaches that reflect realistic constraints and advertising effectiveness.

In Chapter 2, the deterministic algorithms were presented to solve the display ads problem in adversarial order, Then, in Chapter 3, we provided the stochastic online algorithm with scenario-based stochastic programming and Benders decomposition to solve the display ads problem in probabilistic orders. Finally, in Chapter 4, we introduced the online banner advertisement scheduling models by considering critical factors influencing advertising effectiveness and the heuristic approach to obtaining good quality solutions efficiently, even for large data sets. The algorithms the dissertation designed can offer important insights into the online advertisement assignment problem. Next, we summarize our studies presented in Chapters 2, 3, and 4 and future research directions.

In Chapter 2, we considered the display ads problem in adversarial order, which is a generalization of the edge-weighted and capacitated online bipartite matching problem. The objective of this problem is to maximize the total weight of edges matched while considering the strict capacity constraint. The strict capacity constraint reflects realistic situations. It is difficult to solve the online version of the problem with the strict capacity constraint, but we presented a deterministic algorithm with worst-case guarantees. We also proved upper bounds on the competitive ratio of any deterministic algorithms. From the results, we derived that the deterministic algorithm may be a near-optimal algorithm according to the capacity and weight range.

In Chapter 3, we covered the display ads problem in probabilistic orders. The problem in probabilistic orders considered more realistic situation in which the company can stochastically estimate the input sequence by using historical data in real problems although information on weights between edges is revealed online. For the probabilistic orders (*known IID* and *random permutation*), we proposed the stochastic online algorithm with scenario-based stochastic programming and Benders decomposition to solve the problem. The stochastic online algorithms provided better performances than the primal-dual algorithm did through the numerical experiments of the two probabilistic orders. Hence, the solution methodologies provided competitive and realistic solutions in the real-time environment of the assignment problem and are expected to be useful for publishers who place online advertisements on their websites.

There are several future research on the display ads problem covered in Chapters 2 and 3. A possible direction is to design new algorithms that show better performances in terms of the optimality gap and computation time. In addition, it will be useful or meaningful to analyze online advertising assignment problems that have more uncertain structure. For example, both sides of the nodes (advertisements and slots) are revealed online or the weights of all edges are uncertain, not fixed.

In Chapter 4, we considered the online banner advertisement scheduling models to maximize advertising effectiveness. The study contributed to the literature on online banner advertisement scheduling models by considering critical factors influencing advertising effectiveness. In particular, the degree to competition has not yet been discussed extensively in online banner advertisement scheduling. The expected CTR function was devised to reflect the factors in the model and used as an objective function of the model in terms of measuring advertising effectiveness. Because the model we devised was an integer programming model with a non-linear and non-convex objective function, we presented the heuristic algorithm to find valid lower and upper bounds by using the properties of the optimization model. We analyzed the performances of the heuristic by using the small, large, and standard data set. We observed that the heuristic algorithm obtained competitive solutions efficiently even for large data sets.

Some challenging considerations are relevant for future research in Chapter 4. We used the expected CTR values to measure advertising effectiveness. There needs to be future research on validating the similarity (or correlation) between predicted and actual values. The model proposed and described in this study features a variety of parameters related to the effectiveness of advertising. The value of the parameters should be accurately estimated and be dynamically reflected in the model over time. In addition, although the proposed heuristic approach was efficient, it included binary integer programming, which is an NP-hard problem. Therefore, for future research, other methodologies, such as linearization techniques or meta-heuristics, should be considered to solve the non-linear model efficiently.

Appendices

A Initial Sequence of the Hybrid Tabu Search

Phase 1. Calculate R_{it} for each advertisement displayed in each time slot:

$$R_{it} = \alpha_i \times (s_i)^{\beta_{it}}, \ \forall i, t.$$

R_{it} is defined as the expected CTR value without competition effects for ad i displayed in time slot t.

Phase 2. Generate an initial sequence S by N advertisements and T slots in descending order according to R_{it}.

	$R_{i_1 t_1} \geq$	$R_{i_2 t_2} \geq$	$R_{i_3 t_3} \geq$	$\cdots$	$\geq R_{i_{NT} t_{NT}}$
sequence S	(i_1, t_1)	(i_2, t_2)	(i_3, t_3)	$\cdots$	(i_{NT}, t_{NT})
	1^{st} cell	2^{nd} cell	3^{rd} cell	$\cdots$	NT^{th} cell

B Procedure of the Hybrid Tabu Search

Algorithm B.1: Hybrid tabu search

```
Input  : S (initial sequence), H, N, T, s_i (∀i), L_i (∀i)
initialization: CTR ← 0; tabuSEQ ← ∅; num1 ← random(0,1)*;
 SEQ_sol ← S; SEQ ← S;
CTR ← Algorithm B.2(SEQ) ;
while time limit is not reached do
    k ← 1;
    while k < N × T do
        SEQ_k ← sequence by switching the k^th and k+1^th cells of SEQ;
        if SEQ_k ∈ tabuSEQ then
            CTR_k ← 0; k ← k + 1; continue;
        else
            CTR_k ← Algorithm B.2(SEQ_k); k ← k + 1;
        end
    end
    k ← 1; CTR_temp ← 0; temp ← 0;
    while k < N × T do
        if CTR_temp < CTR_k then
            temp ← k;
        end
        k ← k + 1;
    end
    if CTR < CTR_temp then
        CTR ← CTR_temp;
        SEQ_sol ← SEQ_temp;
    end
    tabuSEQ ← tabuSEQ ∪ SEQ_temp;
    SEQ ← SEQ_temp;
end
Output: SEQ_sol; CTR
```

*$random(0, 1)$: an arbitrary constant between 0 and 1

Algorithm B.2: Assigning advertisements to time slots using SEQ

Input : SEQ (sequence)

initialization: $H_t \leftarrow H\ \forall t$; $x_{it} \leftarrow 0\ \forall i, t$; $r_i \leftarrow random(0, 1)^*\ \forall i$;

$\overline{\mathcal{X}} \leftarrow \emptyset$, $t \leftarrow 1$;

\# assigning advertisements to time slots;

$p \leftarrow 1$;

while $p \leq N \times T$ **do**

  $i(t) \leftarrow i_p(t_p)$ in the p^{th} cell of SEQ;

  if $r_i < num1$ **then**

    $p \leftarrow p + 1$; **continue**;

  end

  if $s_i \leq H_t$ **then**

    $x_{it} \leftarrow 1$; $H_t \leftarrow H_t - s_i$; $p \leftarrow p + 1$;

  end

end

\# Checking the constraint and updating the objective values;

$CTR = \sum_{i=1}^{N} \sum_{t=1}^{T} \pi_{it} x_{it}$;

$b \leftarrow 1$;

while $b \leq N$ **do**

  if $L_b \leq \sum_{t=1}^{T} x_{bt}$ or $\sum_{t=1}^{T} x_{bt} = 0$ **then**

    continue;

  else

    $CTR = 0$; **break**;

  end

end

Output: CTR

$^*random(0, 1)$: an arbitrary constant between 0 and 1

C Small Example of the Hybrid Tabu Search

For a small problem, we show how a sequence can be translated into a solution. The height of a banner is 10 and the total number of slots is 3. $num1$ is set to 0. We assume to let the size, lower bound on display frequency, and values of R_{it} for each advertisement be as follows:

Ad i	s_i	L_i	r_i	R_{i1}	R_{i2}	R_{i3}
1	3	0	0.5	10.2	11.4	12.2
2	5	0	0.5	12.1	12.3	14.4
3	2	0	0.5	8.9	8.7	7.9
4	3	0	0.5	9.6	8.4	10.7
5	4	0	0.5	12.4	11.1	10.4

We generate an initial sequence (S) in descending order according to the values of R_{it}:

	R_{23} $\geq$	R_{51} $\geq$	R_{22} $\geq$	R_{13} $\geq$	$\cdots$	R_{42} $\geq$	R_{33}
sequence S	(2, 3)	(5, 1)	(2, 2)	(1, 3)	$\cdots$	(4, 2)	(3, 3)

According to sequence S, we assign advertisements to time slots as follows:

1. The first cell in sequence S is $(i, t) = (2, 3)$ and r_2 is not less than $num1$. Assign ad 2 in slot 3.

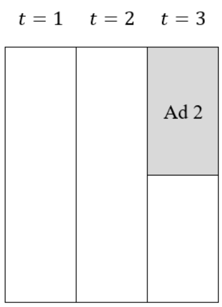

Figure C.1: Assignment result from the first cell

2. The second cell in sequence S is $(i, t) = (5, 1)$ and r_5 is not less than $num1$. Assign ad 5 in slot 1.

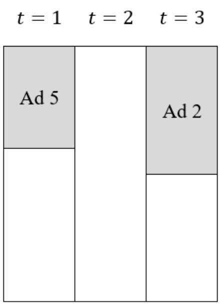

Figure C.2: Assignment result from the second cell

3. The third cell in sequence S is $(i, t) = (2, 2)$ and r_2 is not less than $num1$. Assign ad 2 in slot 2.

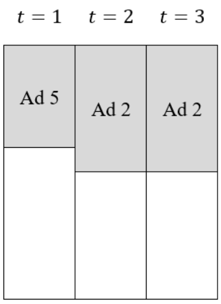

Figure C.3: Assignment result from the third cell

If there is no room to assign ad i in slot t or r_i is less than $num1$, we do not assign ad i in slot t. According to the sequence, we complete an assignment as follows:

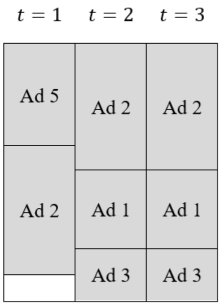

Figure C.4: Final assignment result

If the assignment satisfies Constraint (4.6), the assignment can be a feasible solution. During iterations of the tabu search, we generate new sequences by switching two consecutive cells of the current sequence. We conduct new assignments using new sequences until the tabu search is terminated.

D Linearization Technique of Bilinear Form in $\mathbb{R}^2$

The objective function for online banner advertisement scheduling is as follows:

$$\max \sum_{i=1}^{N} \sum_{t=1}^{T} \alpha_i (s_i)^{\beta_{it}} x_{it} \prod_{\forall j \neq i,} (1 - x_{jt} + s_j x_{jt})^{\delta_{ij}}$$

We can consider this form as a bilinear term (ab) in $\mathbb{R}^2$ $\left(a = \alpha_i (s_i)^{\beta_{it}} x_{it} \text{ and } b = \prod_{\forall j \neq i,} (1 - x_{jt} + s_j x_{jt})^{\delta_{ij}}\right)$. If we know lower (upper) bounds of the variables a and b, the bilinear term can be reformulated as a linearized form [69]. Let $a^L (a^U)$ and $b^L (b^U)$ be the lower (upper) bounds of the variables a and b, respectively. Over the rectangular domain $[a^L,\ a^U] \times [b^L,\ b^U]$, the convex and concave envelopes of ab can be defined as follows [8, 9]:

[Convex envelope (Convex hull)]

$$max\{a^L b + b^L a - a^L b^L, a^U b + b^U a - a^U b^U\}$$

[Concave envelope (Concave hull)]

$$min\{a^U b + b^L a - a^U b^L, a^L b + b^U a - a^L b^U\}$$

Given the two envelopes, the inequalities below are satisfied for all $(a, b) \in [a^L,\ a^U] \times [b^L,\ b^U]$:

$$max\{a^L b + b^L a - a^L b^L, a^U b + b^U a - a^U b^U\} \leq ab$$

$$min\{a^U b + b^L a - a^U b^L, a^L b + b^U a - a^L b^U\} \geq ab$$

If we introduce a new variable e that is replaced by the bilinear term ab, the four constraints below are added in the optimization problems:

$$a^L b + b^L a - a^L b^L \leq e$$

$$a^U b + b^U a - a^U b^U \leq e$$

$$a^U b + b^L a - a^U b^L \geq e$$

$$a^L b + b^U a - a^L b^U \geq e$$

The bilinear term ab can be linearzed by using the four constraints with a new variable ($e = ab$) over the rectangular domain $[a^L,\ a^U] \times [b^L,\ b^U]$.

Bibliography

[1] M. Adler, P. B. Gibbons, and Y. Matias, *Scheduling space-sharing for internet advertising*, Journal of Scheduling, 5 (2002), pp. 103–119.

[2] C. C. Aggarwal, J. L. Wolf, and S. Y. Philip, *A framework for the optimizing of www advertising*, in Trends in Distributed Systems for Electronic Commerce, Springer, 1998, pp. 1–10.

[3] G. Aggarwal, G. Goel, C. Karande, and A. Mehta, *Online vertex-weighted bipartite matching and single-bid budgeted allocations*, in Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms, SIAM, 2011, pp. 1253–1264.

[4] S. Agrawal, Z. Wang, and Y. Ye, *A dynamic near-optimal algorithm for online linear programming*, Operations Research, 62 (2014), pp. 876–890.

[5] W. Agyei, *Mathematical model for television commercial allocation problem*, International Journal of Scientific & Technology Research, 7 (2018), pp. 91–95.

[6] M. T. Ahmed and C. Kwon, *Pricing game of online display advertisement publishers*, European Journal of Operational Research, 219 (2012), pp. 477–487.

[7] ——, *Optimal contract-sizing in online display advertising for publishers with regret considerations*, Omega, 42 (2014), pp. 201–212.

[8] F. A. Al-Khayyal, *Jointly constrained bilinear programs and related problems: An overview*, Computers & Mathematics with Applications, 19 (1990), pp. 53–62.

[9] F. A. Al-Khayyal and J. E. Falk, *Jointly constrained biconvex programming*, Mathematics of Operations Research, 8 (1983), pp. 273–286.

[10] S. Albers and S. Leonardi, *Online algorithms*, ACM Computing surveys, 31 (1999), pp. 10–1145.

[11] A. Amiri and S. Menon, *Scheduling web banner advertisements with multiple display frequencies*, IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 36 (2006), pp. 245–251.

[12] J. F. Benders, *Partitioning procedures for solving mixed-variables programming problems*, Computational Management Science, 2 (2005), pp. 3–19.

[13] A. Bernstein, J. Holm, and E. Rotenberg, *Online bipartite matching with amortized o (log 2 n) replacements*, Journal of the ACM (JACM), 66 (2019), pp. 1–23.

[14] A. Bhalgat, J. Feldman, and V. Mirrokni, *Online allocation of display ads with smooth delivery*, in Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, 2012, pp. 1213–1221.

[15] U. Bhaskar, A. Jalal, and R. Vaze, *The adwords problem with strict capacity constraints*, in 36th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2016), Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[16] S. Bhat, M. Bevans, and S. Sengupta, *Measuring users' web activity to evaluate and enhance advertising effectiveness*, Journal of Advertising, 31 (2002), pp. 97–106.

[17] B. Birnbaum and C. Mathieu, *On-line bipartite matching made simple*, Acm Sigact News, 39 (2008), pp. 80–87.

[18] V. Boskamp, A. Knoops, F. Frasincar, and A. Gabor, *Maximizing revenue with allocation of multiple advertisements on a web banner*, Computers & Operations Research, 38 (2011), pp. 1412–1424.

[19] N. Buchbinder, K. Jain, and J. S. Naor, *Online primal-dual algorithms for maximizing ad-auctions revenue*, in European Symposium on Algorithms, Springer, 2007, pp. 253–264.

[20] N. Buchbinder, D. Segev, and Y. Tkach, *Online algorithms for maximum cardinality matching with edge arrivals*, Algorithmica, 81 (2019), pp. 1781–1799.

[21] R. L. Carraway, R. L. Schmidt, and L. R. Weatherford, *An algorithm for maximizing target achievement in the stochastic knapsack problem with normal returns*, Naval Research Logistics (NRL), 40 (1993), pp. 161–173.

[22] P. P. Chan, X. Hu, L. Zhao, D. S. Yeung, D. Liu, and L. Xiao, *Convolutional neural networks based click-through rate prediction with multiple feature sequences.*, in IJCAI, 2018, pp. 2007–2013.

[23] M. Chang, D. S. Hochbaum, Q. Spaen, and M. Velednitsky, *Dispatch: an optimally-competitive algorithm for maximum online perfect bipartite matching with iid arrivals*, in International Workshop on Approximation and Online Algorithms, Springer, 2018, pp. 149–164.

[24] ——, *An optimally-competitive algorithm for maximum online perfect bipartite matching with iid arrivals*, Theory of Computing Systems, (2019), pp. 1–17.

[25] D. Charles, M. Chickering, N. R. Devanur, K. Jain, and M. Sanghi, *Fast algorithms for finding matchings in lopsided bipartite graphs with applications to display ads*, in Proceedings of the 11th ACM conference on Electronic commerce, 2010, pp. 121–128.

[26] X. A. Chen and Z. Wang, *A dynamic learning algorithm for online matching problems with concave returns*, European Journal of Operational Research, 247 (2015), pp. 379–388.

[27] Y. Chen, D. Pavlov, and J. F. Canny, *Large-scale behavioral targeting*, in Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, 2009, pp. 209–218.

[28] H. Choi, C. F. Mela, S. Balseiro, and A. Leary, *Online display advertising markets: A literature review and future directions*, Columbia Business School Research Paper, (2019).

[29] D. F. Ciocan and V. F. Farias, *Dynamic allocation problems with volatile demand*, Mathematics of Operations Research, 37 (2012), pp. 501–525.

[30] M. Corstjens and P. Doyle, *A model for optimizing retail space allocations*, Management Science, 27 (1981), pp. 822–833.

[31] R. C. Curhan, *Shelf space allocation and profit maximization in mass retailing*, Journal of Marketing, 37 (1973), pp. 54–60.

[32] K. Czerniachowska, *Scheduling tv advertisements via genetic algorithm*, European Journal of Industrial Engineering, 13 (2019), pp. 81–116.

[33] M. R. da Silva, R. C. Schouery, and L. L. Pedrosa, *A polynomial-time approximation scheme for the maxspace advertisement problem*, Electronic Notes in Theoretical Computer Science, 346 (2019), pp. 699–710.

[34] M. Dawande, S. Kumar, and C. Sriskandarajah, *Performance bounds of algorithms for scheduling advertisements on a web page*, Journal of Scheduling, 6 (2003), pp. 373–394.

[35] J. Deane, *Hybrid genetic algorithm and augmented neural network application for solving the online advertisement scheduling problem with contextual targeting*, Expert Systems with Applications, 39 (2012), pp. 5168–5177.

[36] J. Deane and A. Agarwal, *Scheduling online advertisements to maximize revenue under non-linear pricing*, Journal of Computer Information Systems, 53 (2012), pp. 85–92.

[37] ——, *Scheduling online advertisements to maximize revenue under variable display frequency*, Omega, 40 (2012), pp. 562–570.

[38] J. Deane and P. Pathak, *Ontological analysis of web surf history to maximize the click-through probability of web advertisements*, Decision Support Systems, 47 (2009), pp. 364–373.

[39] P. Desmet and V. Renaudin, *Estimation of product category sales responsiveness to allocated shelf space*, International Journal of Research in Marketing, 15 (1998), pp. 443–457.

[40] N. R. Devanur and T. P. Hayes, *The adwords problem: online keyword matching with budgeted bidders under random permutations*, in Proceedings of the 10th ACM conference on Electronic commerce, 2009, pp. 71–78.

[41] N. R. Devanur, K. Jain, B. Sivan, and C. A. Wilkens, *Near optimal online algorithms and fast approximation algorithms for resource allocation problems*, Journal of the ACM (JACM), 66 (2019), pp. 1–41.

[42] A. Deza, K. Huang, and M. R. Metel, *Chance constrained optimization for targeted internet advertising*, Omega, 53 (2015), pp. 90–96.

[43] X. Dreze, S. J. Hoch, and M. E. Purk, *Shelf management and space elasticity*, Journal of retailing, 70 (1994), pp. 301–326.

[44] G. Einziger, C. F. Chiasserini, and F. Malandrino, *Scheduling advertisement delivery in vehicular networks*, IEEE Transactions on Mobile Computing, 17 (2018), pp. 2882–2897.

[45] M. Eisend, *Shelf space elasticity: A meta-analysis*, Journal of Retailing, 90 (2014), pp. 168–181.

[46] S. A. Erdogan, A. Gose, and B. T. Denton, *Online appointment sequencing and scheduling*, IIE Transactions, 47 (2015), pp. 1267–1286.

[47] U. Feige, *Tighter bounds for online bipartite matching*, in Building Bridges II, Springer, 2019, pp. 235–255.

[48] J. Feldman, M. Henzinger, N. Korula, V. S. Mirrokni, and C. Stein, *Online stochastic packing applied to display ad allocation*, in European Symposium on Algorithms, Springer, 2010, pp. 182–194.

[49] J. Feldman, N. Korula, V. Mirrokni, S. Muthukrishnan, and M. Pál, *Online ad assignment with free disposal*, in International workshop on internet and network economics, Springer, 2009, pp. 374–385.

[50] J. Feldman, A. Mehta, V. Mirrokni, and S. Muthukrishnan, *Online stochastic matching: Beating 1-1/e*, in 2009 50th Annual IEEE Symposium on Foundations of Computer Science, IEEE, 2009, pp. 117–126.

[51] A. Fiat and G. J. Woeginger, *Online algorithms: The state of the art*, vol. 1442, Springer, 1998.

[52] A. Freund and J. S. Naor, *Approximating the advertisement placement problem*, Journal of Scheduling, 7 (2004), pp. 365–374.

[53] K. Fridgeirsdottir and S. Najafi-Asadolahi, *Cost-per-impression pricing for display advertising*, Operations Research, 66 (2018), pp. 653–672.

[54] I. Gamzu and I. Koutsopoulos, *Advertisement allocation and mechanism design in native stream advertising*, in International Conference on Complex Networks and their Applications, Springer, 2018, pp. 197–210.

[55] R. Gatarski, *Breed better banners: Design automation through on-line interaction*, Journal of Interactive Marketing, 16 (2002), pp. 2–13.

[56] F. Glover and M. Laguna, *Tabu search*, in Handbook of combinatorial optimization, Springer, 1998, pp. 2093–2229.

[57] G. Goel and A. Mehta, *Online budgeted matching in random input models with applications to adwords.*, in SODA, vol. 8, 2008, pp. 982–991.

[58] R. Gopal, X. Li, and R. Sankaranarayanan, *Online keyword based advertising: Impact of ad impressions on own-channel and cross-channel click-through rates*, Decision Support Systems, 52 (2011), pp. 1–8.

[59] A. Gupta, S. Aggarwal, A. Kaul, and P. Jha, *Optimal placement of advertisements on a pixelated web banner*, Journal of Information and Optimization Sciences, 37 (2016), pp. 693–716.

[60] A. Gupta, P. Manik, S. Aggarwal, and P. Jha, *Optimal advertisement planning on online news media*, in Proceedings of International Congress on Productivity, Quality, Reliability, Optimization and Modelling, vol. 2, 2011, pp. 963–978.

[61] B. Haeupler, V. S. Mirrokni, and M. Zadimoghaddam, *Online stochastic weighted matching: Improved approximation algorithms*, in International workshop on internet and network economics, Springer, 2011, pp. 170–181.

[62] S. Henningsen, R. Heuke, and M. Clement, *Determinants of advertising effectiveness: The development of an international advertising elasticity database and a meta-analysis*, Business Research, 4 (2011), pp. 193–239.

[63] A. Hertz, R. Montagné, and F. Gagnon, *Online algorithms for the maximum k-colorable subgraph problem*, Computers & Operations Research, 91 (2018), pp. 209–224.

[64] Z. Huang, B. Peng, Z. G. Tang, R. Tao, X. Wu, and Y. Zhang, *Tight competitive ratios of classic matching algorithms in the fully online model*, in Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2019, pp. 2875–2886.

[65] Z. Huang, Z. G. Tang, X. Wu, and Y. Zhang, *Online vertex-weighted bipartite matching: Beating 1-1/e with random arrivals*, arXiv preprint arXiv:1804.07458, (2018).

[66] IAB, *Iab internet advertising revenue report 2016 full-year results*, tech. rep., Technical Report. Market research report, Interactive Advertising Bureau, 2017.

[67] ——, *Iab internet advertising revenue report 2017 full-year results*, tech. rep., Technical Report. Market research report, Interactive Advertising Bureau, 2018.

[68] ——, *Iab internet advertising revenue report 2018 full-year results*, tech. rep., Technical Report. Market research report, Interactive Advertising Bureau, 2019.

[69] J. Irion, J. Lu, F. Al-Khayyal, and Y. Tsao, *A piecewise linearization framework for retail shelf space management models*, European Journal of Operational Research, 222 (2012), pp. 122–136.

[70] P. Jaillet and X. Lu, *Near-optimal online algorithms for dynamic resource allocation problems*, arXiv preprint arXiv:1208.2596, (2012).

[71] ——, *Online stochastic matching: New algorithms with better bounds*, Mathematics of Operations Research, 39 (2014), pp. 624–646.

[72] P. Jaillet and M. R. Wagner, *Online Optimization*, Springer Publishing Company, Incorporated, 2012.

[73] K. Jea, J. Wang, and C. Hsu, *Two-agent advertisement scheduling on physical books to maximize the total profit*, Asia-Pacific Journal of Operational Research (APJOR), 36 (2019), pp. 1–24.

[74] W. Ji, Y. Chen, M. Chen, B. Chen, Y. Chen, and S. Kung, *Profit maximization through online advertising scheduling for a wireless video broadcast network*, IEEE Transactions on Mobile Computing, 15 (2015), pp. 2064–2079.

[75] Y. Juan, Y. Zhuang, W.-S. Chin, and C.-J. Lin, *Field-aware factorization machines for ctr prediction*, in Proceedings of the 10th ACM Conference on Recommender Systems, 2016, pp. 43–50.

[76] B. Kalyanasundaram and K. R. Pruhs, *An optimal deterministic algorithm for online b-matching*, Theoretical Computer Science, 233 (2000), pp. 319–325.

[77] K. Kang, K. Altinkemer, and I. Hur, *Mobile coupons delivery problem: Postponable online multi-constraint knapsack*, Decision Support Systems, 116 (2019), pp. 13–25.

[78] C. Karande, A. Mehta, and P. Tripathi, *Online bipartite matching with unknown distributions*, in Proceedings of the forty-third annual ACM symposium on Theory of computing, 2011, pp. 587–596.

[79] R. M. Karp, *An introduction to randomized algorithms*, Discrete Applied Mathematics, 34 (1991), pp. 165–201.

[80] ——, *On-line algorithms versus off-line algorithms: How much is it worth to know the future?*, in IFIP congress (1), vol. 12, 1992, pp. 416–429.

[81] R. M. Karp, U. V. Vazirani, and V. V. Vazirani, *An optimal algorithm for on-line bipartite matching*, in Proceedings of the twenty-second annual ACM symposium on Theory of computing, 1990, pp. 352–358.

[82] A. Kaul, S. Aggarwal, A. Gupta, N. Dayama, M. Krishnamoorthy, and P. Jha, *Optimal advertising on a two-dimensional web banner*, International Journal of System Assurance Engineering and Management, 9 (2018), pp. 306–311.

[83] T. Kesselheim, K. Radke, A. Tönnis, and B. Vöcking, *An optimal online algorithm for weighted bipartite matching and extensions to combinatorial auctions*, in European symposium on algorithms, Springer, 2013, pp. 589–600.

[84] A. Khatibi, G. Baharian, E. R. Kone, and S. H. Jacobson, *The sequential stochastic assignment problem with random success rates*, IIE Transactions, 46 (2014), pp. 1169–1180.

[85] A. Knoops, V. Boskamp, A. Wojciechowski, and F. Frasincar, *Single pattern generating heuristics for pixel advertisements*, in International Conference on Web Information Systems Engineering, Springer, 2009, pp. 415–428.

[86] N. Korula and M. Pál, *Algorithms for secretary problems on graphs and hypergraphs*, in International Colloquium on Automata, Languages, and Programming, Springer, 2009, pp. 508–520.

[87] S. Kumar, M. Dawande, and V. Mookerjee, *Optimal scheduling and placement of internet banner advertisements*, IEEE Transactions on Knowledge and Data Engineering, 19 (2007), pp. 1571–1584.

[88] S. Kumar, V. S. Jacob, and C. Sriskandarajah, *Scheduling advertisements on a web page to maximize revenue*, European journal of operational research, 173 (2006), pp. 1067–1089.

[89] W. Lee, *Space management in retail stores and implications to agriculture*, Marketing Keys to Profits in the 1960's, (1961), pp. 523–533.

[90] A. Legrain and P. Jaillet, *A stochastic algorithm for online bipartite resource allocation problems*, Computers & Operations Research, 75 (2016), pp. 28–37.

[91] M. A. Lejeune and J. Turner, *Planning online advertising using gini indices*, Operations Research, 67 (2019), pp. 1222–1245.

[92] H. Li and J. L. Bukovac, *Cognitive impact of banner ad characteristics: An experimental study*, Journalism & Mass Communication Quarterly, 76 (1999), pp. 341–353.

[93] Y. Lin and Y. Chen, *Effects of ad types, positions, animation lengths, and exposure times on the click-through rate of animated online advertisings*, Computers & Industrial Engineering, 57 (2009), pp. 580–591.

[94] M. Mahdian and Q. Yan, *Online bipartite matching with random arrivals: an approach based on strongly factor-revealing lps*, in Proceedings of the forty-third annual ACM symposium on Theory of computing, 2011, pp. 597–606.

[95] P. Manik, A. Gupta, P. Jha, and K. Govindan, *A goal programming model for selection and scheduling of advertisements on online news media*, Asia-Pacific Journal of Operational Research, 33 (2016), p. 1650012.

[96] V. H. Manshadi, S. O. Gharan, and A. Saberi, *Online stochastic matching: Online actions based on offline statistics*, Mathematics of Operations Research, 37 (2012), pp. 559–573.

[97] J. Marszałkowski and M. Drozdowski, *Optimization of column width in website layout for advertisement fit*, European Journal of Operational Research, 226 (2013), pp. 592–601.

[98] M. McCandless, *Web advertising*, IEEE Intelligent Systems and their Applications, 13 (1998), pp. 8–9.

[99] A. Mehta, *Online matching and ad allocation*, Foundations and Trends® in Theoretical Computer Science, 8 (2013), pp. 265–368.

[100] A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani, *Adwords and generalized online matching*, Journal of the ACM (JACM), 54 (2007), pp. 22–es.

[101] V. S. Mirrokni, S. O. Gharan, and M. Zadimoghaddam, *Simultaneous approximations for adversarial and stochastic online budgeted allocation*, in Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms, SIAM, 2012, pp. 1690–1701.

[102] L. Mitten, *Composition principles for synthesis of optimal multistage processes*, Operations Research, 12 (1964), pp. 610–619.

[103] T. L. Morin, *Monotonicity and the principle of optimality*, Journal of Mathematical Analysis and Applications, 88 (1982), pp. 665–674.

[104] D. Mougouei, D. M. Powers, and A. Moeini, *An integer linear programming model for binary knapsack problem with dependent item values*, in Australasian Joint Conference on Artificial Intelligence, Springer, 2017, pp. 144–154.

[105] J. Naor and D. Wajc, *Near-optimum online ad allocation for targeted advertising*, ACM Transactions on Economics and Computation (TEAC), 6 (2018), pp. 1–20.

[106] T. P. Novak and D. L. Hoffman, *New metrics for new media: Toward the development of web measurement standards.*, World Wide Web Journal, 2 (1997), pp. 213–246.

[107] V. Palade and S. Banerjee, *Web ad-slot offline scheduling using an ant colony algorithm*, in 2011 10th International Conference on Machine Learning and Applications and Workshops, vol. 2, IEEE, 2011, pp. 263–268.

[108] S. Purnamawati, E. Nababan, B. Tsani, R. Taqyuddin, and R. Rahmat, *Advertisement scheduling on commercial radio station using genetics algorithm*, in Journal of Physics: Conference Series, vol. 978, IOP Publishing, 2018.

[109] X. Qiu and H. Y. Lau, *An ais-based hybrid algorithm with pdrs for multi-objective dynamic online job shop scheduling problem*, Applied Soft Computing, 13 (2013), pp. 1340–1351.

[110] N. Rajaraman and R. Vaze, *Submodular maximization under a matroid constraint: Asking more from an old friend, the greedy algorithm*, arXiv preprint arXiv:1810.12861, (2018).

[111] J. Ren, J. V. Nickerson, W. Mason, Y. Sakamoto, and B. Graber, *Increasing the crowd's capacity to create: how alternative generation affects the diversity, relevance and effectiveness of generated ads*, Decision Support Systems, 65 (2014), pp. 28–39.

[112] A. J. Rethans, J. L. Swasy, and L. J. Marks, *Effects of television commercial repetition, receiver knowledge, and commercial length: A test of the two-factor model*, Journal of Marketing Research, 23 (1986), pp. 50–61.

[113] G. Rosenkrans, *Maximizing user interactivity through banner ad design*, Journal of Promotion Management, 16 (2010), pp. 265–287.

[114] K. Schaal and A. Hübner, *When does cross-space elasticity matter in shelf-space planning? a decision analytics approach*, Omega, 80 (2018), pp. 135–152.

[115] A. Shapiro and A. Philpott, *A tutorial on stochastic programming*, Manuscript. Available at www2. isye. gatech. edu/ashapiro/publications. html, 17 (2007).

[116] L. Shi and B. Li, *Predict the click-through rate and average cost per click for keywords using machine learning methodologies*, in Proceedings of the International Conference on Industrial Engineering and Operations Management-Detroit, Michigan, USA, 2016.

[117] X. Sun, J. Zhang, and J. Zhang, *Near optimal algorithms for online weighted bipartite matching in adversary model*, Journal of Combinatorial Optimization, 34 (2017), pp. 689–705.

[118] H. Ting and X. Xiang, *Near optimal algorithms for online maximum edge-weighted b-matching and two-sided vertex-weighted b-matching*, Theoretical Computer Science, 607 (2015), pp. 247–256.

[119] A. Torrico, S. Ahmed, and A. Toriello, *A polyhedral approach to online bipartite matching*, Mathematical Programming, 172 (2018), pp. 443–465.

[120] D. Vakratsas and T. Ambler, *How advertising works: what do we really know?*, Journal of marketing, 63 (1999), pp. 26–43.

[121] S. Wu, P. Wei, and J. Chen, *Influential factors and relational structure of internet banner advertising in the tourism industry*, Tourism Management, 29 (2008), pp. 221–236.

[122] R. R. Yager, *Intelligent agents for world wide web advertising decisions*, International Journal of Intelligent Systems, 12 (1997), pp. 379–390.

[123] W. Zhang, T. Du, and J. Wang, *Deep learning over multi-field categorical data*, in European conference on information retrieval, Springer, 2016, pp. 45–57.

국문초록

온라인 커뮤니티의 급격한 성장에 따라, 많은 회사들이 온라인 광고에 관심을 기울이고 있다. 온라인 광고의 장점으로는 추적 가능성, 비용 효과성, 도달 가능성, 상호작용성 등이 있다. 온라인에 기반을 두는 회사들은 잘 짜여진 온라인 광고 할당결정에 관심을 두고 있고, 이는 광고 수익과 연관될 수 있다. 따라서 온라인 광고 관리자는 수익을 극대화 할 수 있는 온라인 광고 할당 의사 결정 프로세스를 개발하여야 한다.

본 논문에서는 현실적인 제약을 고려한 온라인 광고 할당 문제들을 제안한다. 본 논문에서 다루는 문제는 (1) adversarial 순서로 진행하는 디스플레이 애드문제, (2) probabilistic 순서로 진행하는 디스플레이 애드문제 그리고 (3) 광고효과를 위한 온라인 배너 광고 일정계획이다. 이전에 제안되었던 광고 할당 문제들과 달리, 본 논문에서 제안한 문제들은 현실적인 제약과 광고효과를 반영하는 실용적인 접근 방식이다. 또한 제안하는 알고리즘은 온라인 광고 할당 문제의 운영관리에 대한 통찰력을 제공한다.

1장에서는 온라인 광고 할당 문제에 대한 문제해결 방법론에 대해 간단히 소개한다. 더불어 연구의 기여와 개요도 제공된다. 2장에서는 adversarial 순서로 진행하는 디스플레이 애드문제를 제안한다. worst-case를 보장하는 결정론적 알고리즘을 설계하고, 이들의 competitive ratio를 증명한다. 더불어 문제의 상한도 입증된다. 3장에서는 probabilistic 순서로 진행하는 디스플레이 애드문제를 제안한다. 시나리오 기반의 확률론적 온라인 알고리즘과 Benders 분해방법을 혼합한 추계 온라인 알고리즘을 제시한다. 4장에서는 광고효과를 위한 온라인 배너 광고 일정계획을 설계한다. 또한, 모델의 유효한 상한과 하한을 효율적으로 얻는 데 사용되는 문제해결 방법론을 제안한다. 5장에서는 본 논문의 결론과 향후 연구를 위한 방향을 제공한다.

본 논문에서 제안하는 문제해결 방법론은 학술 및 산업 분야 모두 의미가 있다. 수치

실험을 통해 문제해결 접근 방식이 문제를 효율적이고 효과적으로 해결할 수 있음을 보인다. 이는 온라인 광고 관리자가 본 논문에서 제안하는 문제와 문제해결 방법론을 통해 온라인 광고 할당관련 의사결정을 진행하는 데 있어 도움이 될 것으로 기대한다.

주요어: 온라인 광고 할당문제, 디스플레이 애드문제, 온라인 배너 광고 스케줄링, 온라인 알고리즘, 추계 계획법, 광고효과

학번: 2014 - 22636

감사의 글

아무것도 모른 채 더 넓은 환경에서 공부하겠다는 일념으로 관악에 온 지 어느덧 10년이 넘었습니다. 학부 4년 반 그리고 대학원 6년 동안의 시간을 돌이켜보면서 여러 가지 기억과 감정들이 주마등처럼 지나갑니다. 그 중 가장 먼저 들었던 생각은 지치고 포기하고 싶었던 시간이 찾아올 때마다 많은 분들의 도움으로 지금 이 순간까지 올 수 있었다는 사실입니다. 감사한 마음을 이 글로 표현하기에는 한없이 부족해 부끄럽지만, 그럼에도 한 자 한 자 감사한 마음을 적어보고자 합니다.

대학원 생활에 의도치 않은 변화가 찾아왔을 때, 저에게 한 번 더 기회를 주신 문일경 교수님께 깊이 감사드립니다. 연구를 진행하면서 좋은 결실을 맺을 수 있도록 항상 바른 길로 지도해 주셨고, 뜻한대로 풀리지 않을 때에도 어느 누구보다 큰 격려와 응원을 보내주셨습니다. 교수님께서 보여주신 가르침 덕분에 저는 학문적으로나 인성적으로나 한층 더 성장할 수 있었습니다. 언제나 책임감을 보여주시고 아끼지 않는 지원과 관심을 주시는 모습에 연구실 생활에 정진할 수 있었고 많은 것을 배울 수 있었습니다. 교수님의 모습을 거울로 삼아 저 또한 타인에게 그러한 사람이 될 수 있도록 노력하겠습니다. 존경합니다, 교수님.

본 학위 논문의 심사를 맡아 주신 이경식 교수님, 박건수 교수님, 이영훈 교수님 그리고 정태수 교수님께 감사드립니다. 교수님들께서 주신 아낌없는 조언은 레이싱 끝자락에 안주할 뻔했던 저에게 큰 깨달음을 주었고, 이에 결승선까지 잘 도착할 수 있었습니다. 심사위원장을 맡아 주신 이경식 교수님, 교수님의 수업을 통해 최적화라는 분야에 다시 관심을 가질 수 있었습니다. 아직 부족하지만 교수님께서 말씀해주신 조언들을 바탕으로 연구자로 살아가는데 있어, 제 스스로를 조금 더 객관적으로 바라보는 냉철함을 키울 수 있도록 더욱 노력하겠습니다. 박건수 교수님, 언제나 친근하고 따뜻하게 격려해주시면서도 지도해주실 때만큼은 연구의 방향성에 대해 날카로운 조

언을 해주셨습니다. 논문을 마무리하는 데 있어 제가 올바른 방향을 찾아갈 수 있게 해 주심에 감사드립니다. 이영훈 교수님, 심사 때 말씀해주신 연구자가 지녀야할 자세 그리고 넓고 새로운 시각에서 연구를 접근할 수 있어야 한다는 말씀은 저에게 많은 자극이 되었습니다. 연구자로서의 삶뿐만 아니라 제가 살아가는 데 있어 교수님께서 주신 조언은 항상 잊지 않고 마음에 새기겠습니다. 정태수 교수님, 교수님의 꼼꼼한 지도와 조언 그리고 따뜻한 말들이 저에게 큰 힘이 되었습니다. 저도 누군가에게 버팀목이 될 수 있는 사람이 되고 싶습니다. 보내주신 격려에 감사드립니다. 더불어, 10년 간 전공 지식뿐만 아니라 어른의 지혜라는 것을 가르쳐 주신 산업공학과 모든 교수님들께도 감사드립니다.

저에게 공급망관리 연구실 동료들은 함께 보낸 시간만큼이나 큰 마음으로 감사함을 표현해야 하지만 그렇지 못해서 더욱 미안한 사람들입니다. 힘들고 어려울 때마다 발 벗고 나서서 응원과 공감을 해주었고 더 나아가 미처 생각하지 못했던 솔루션까지 주는 그런 멋진 동료들입니다. 공급망관리 연구실에 들어오게 되었을 때 방황하던 저를 동료들이 따뜻하게 반겨주었기 때문에 이 순간까지 버틸 수 있었습니다. 부족하지만 이 글을 통해 동료들에게 감사의 마음을 전합니다. 후에 힘든 순간이 찾아오더라도 현명하게 극복하고, 보람찬 그리고 행복한 연구실 생활을 보낼 수 있기를 기원합니다.

학부생활을 마치고, 저에게 찾아온 큰 변화 중 하나였던 대학원 생활을 할 수 있게 도와 주신 서은석 교수님께 감사의 말씀을 전합니다. 비록 짧은 시간이었지만 저와 연구실 동료들에게 주신 따뜻한 마음 잊지 않겠습니다. 그리고 그 기간에 곁에서 생활하며 서로에게 공감하고 의지했던 전략공학 연구실 선배님들과 동료들, 좋은 분들이라 지금까지도 인연을 이어 나갈 수 있었고 덕분에 저에게 큰 힘이 되었습니다. 감사합니다. 또한 학부와 대학원 생활 동안 만난 선배님들, 동기들 그리고 후배님들 소중한 만남이라 생각해 잊지 않고 항상 감사한 마음 새기겠습니다.

박사과정의 마무리를 앞두고 있는 지금 이 순간, 제가 받았던 감사한 마음과 도움을 항상 잊지 않겠습니다. 이 마음을 간직하며 저 또한 누군가에게 그 마음 몇 곱절 더 베풀

수 있는 사람이 되길 다짐해봅니다. 마지막으로 무슨 일이 있어도 항상 저의 편이 되어주고 묵묵히 믿음과 응원을 보내주는 가족들에게 감사드립니다. 강함보다 다정다감의 힘을 보여주고 저에게 정신적 지주가 되어주는 아버지, 사랑하는 마음으로 보듬어 주고 언제든지 기댈 수 있는 버팀목이 되어주는 어머니 그리고 안 그런 척 하지만 뒤에서 항상 응원해주고 걱정해주는 내 동생 향이에게 본 논문을 바칩니다. 앞으로도 자랑스러운 아들 그리고 오빠가 되겠습니다. 사랑합니다.

2020년 6월

김 광 올림